MKW36/35 Reference Manual

Bluetooth® Low Energy System on a chip that supports:
MKW36A512VHT4, MKW36A512VFP4, MKW36Z512VHT4,
MKW36Z512VFP4, MKW35A512VFP4, MKW35Z512VHT4

Document Number: MKW36A512RM
Rev. 5, 07/2018

Preliminary

Contents

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

Chapter 1
About This Manual

Chapter 2
Introduction

Chapter 3
Signal Multiplexing and Pin Assignment

Chapter 4
Chip Configuration

Chapter 5
Memory Map

Chapter 6
Clock Distribution

Chapter 7
Reset and Boot

Chapter 8
Power Management

Chapter 9
Security

Chapter 10
Debug

Chapter 11
Kinetis Flashloader

Chapter 12
Port control and interrupt (PORT)

Chapter 13
System Integration Module (SIM)

Chapter 14
System Mode Controller (SMC)

Chapter 15
Power Management Controller (PMC)

Chapter 16
DCDC Converter (DCDC)

Chapter 17
Low-Leakage Wakeup Unit (LLWU)

Chapter 18
Reset Control Module (RCM)

Chapter 19
Bit Manipulation Engine (BME)

Chapter 20
Miscellaneous Control Module (MCM)

Chapter 21
Micro Trace Buffer (MTB)

Chapter 22
Crossbar Switch Lite (AXBS-Lite)

Chapter 23

Peripheral Bridge (AIPS-Lite)

Chapter 24
Direct Memory Access Multiplexer (DMAMUX)

Chapter 25
DMA Controller Module

Chapter 26
Multipurpose Clock Generator (MCG)

Chapter 27
32 kHz Oscillator (32kRTC)

Chapter 28
Flash Memory Controller (FMC)

Chapter 29
Flash Memory Module (FTFE)

Chapter 30
Analog-to-digital converter (ADC)

Chapter 31
Comparator (CMP)

Chapter 32
Voltage Reference(VREF)

Chapter 33
Timer/PWM Module (TPM)

Chapter 34
Periodic interrupt timer (PIT)

Chapter 35
Low-power timer (LPTMR)

Chapter 36
Real Time Clock (RTC)

Chapter 37
FlexCAN

Chapter 38
Serial Peripheral Interface (SPI)

Chapter 39
Inter-Integrated Circuit (I2C)

Chapter 40
Low Power Universal Asynchronous Receiver/Transmitter (LPUART)

Chapter 41
Carrier Modulator Transmitter (CMT)

Chapter 42
General-purpose input/output (GPIO)

Chapter 43
LP Trusted Cryptography (LTC)

Chapter 44
True Random Number Generator (TRNG)

Chapter 45
2.4 GHz Multi-Protocol Radio

Chapter 1
About This Manual

1.1 Audience

This reference manual is intended for system software and hardware developers and applications programmers who want to develop products with this device. It assumes that the reader understands operating systems, microprocessor system design, and basic principles of software and hardware.

1.2 Organization

This manual has two main sets of chapters.

1. Chapters in the first set contain information that applies to all components on the chip. The *Chip Configuration* chapter contains chip-specific information, which includes the number of module instances on the chip and possible implementation differences between the module instances, such as differences in FIFO depths or the number of channels supported. It may also include functional connections between the module instances and other modules. Read this section first because its content is crucial to understanding the information in other sections of the chapter.
2. Chapters in the second set are organized into functional groupings that detail particular areas of functionality.
   - Examples of these groupings are clocking, timers, and communication interfaces.
   - Each grouping includes chapters that provide a technical description of individual modules. The chapters provide general information about the module, including its signals, registers, and functional description.

NOTE

If there is a conflict between the chip-specific module information and the general module information, the chip-specific information supersedes the general information.

1.3 Register descriptions

Module chapters present register information in:

- Memory maps including:
  - Addresses
  - The name and acronym/abbreviation of each register
  - The width of each register (in bits)
  - Each register's reset value
  - The page number on which each register is described
- Register figures
- Field-description tables
- Associated text

The register figures show the field structure using the conventions in the following figure.

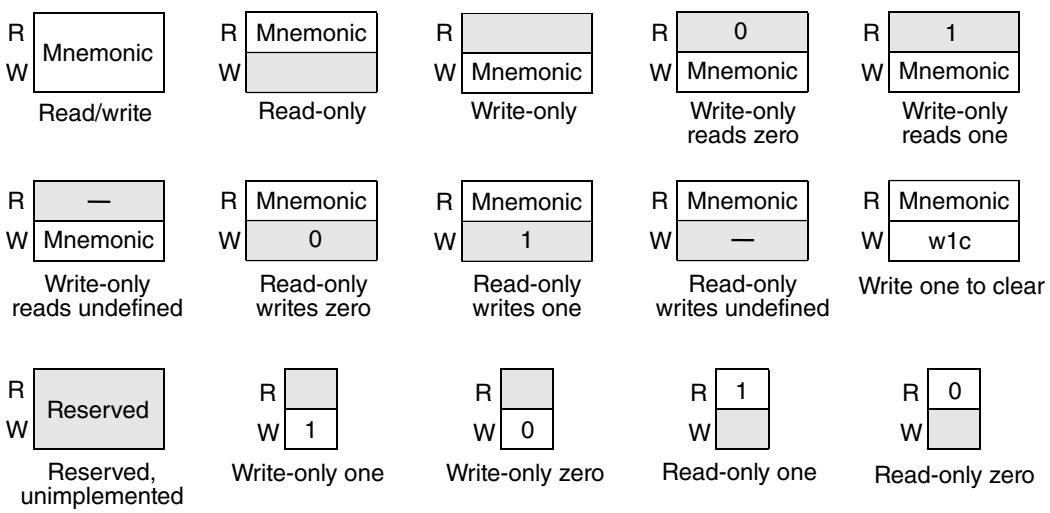


Figure 1-1. Register figure conventions

1.4 Conventions

1.4.1 Numbering systems

The following suffixes identify different numbering systems:

This suffix	Identifies a
b	Binary number. For example, the binary equivalent of the number 5 is written 101b. In some cases, binary numbers are shown with the prefix *0b*.
d	Decimal number. Decimal numbers are followed by this suffix only when the possibility of confusion exists. In general, decimal numbers are shown without a suffix.
h	Hexadecimal number. For example, the hexadecimal equivalent of the number 60 is written 3Ch. In some cases, hexadecimal numbers are shown with the prefix *0x*.

1.4.2 Typographic notation

The following typographic notation is used throughout this document:

Example	Description
placeholder, x	Items in italics are placeholders for information that you provide. Italicized text is also used for the titles of publications and for emphasis. Plain lowercase letters are also used as placeholders for single letters and numbers.
`code`	Fixed-width type indicates text that must be typed exactly as shown. It is used for instruction mnemonics, directives, symbols, subcommands, parameters, and operators. Fixed-width type is also used for example code. Instruction mnemonics and directives in text and tables are shown in all caps; for example, BSR.
SR[SCM]	A mnemonic in brackets represents a named field in a register. This example refers to the Scaling Mode (SCM) field in the Status Register (SR).
REVNO[6:4], XAD[7:0]	Numbers in brackets and separated by a colon represent either: • A subset of a register's named field For example, REVNO[6:4] refers to bits 6–4 that are part of the COREREV field that occupies bits 6–0 of the REVNO register. • A continuous range of individual signals of a bus For example, XAD[7:0] refers to signals 7–0 of the XAD bus.

1.4.3 Special terms

The following terms have special meanings:

Term	Meaning
asserted	Refers to the state of a signal as follows: • An active-high signal is asserted when high (1). • An active-low signal is asserted when low (0).
deasserted	Refers to the state of a signal as follows: • An active-high signal is deasserted when low (0). • An active-low signal is deasserted when high (1). In some cases, deasserted signals are described as *negated*.

Table continues on the next page...

Term	Meaning
reserved	Refers to a memory space, register, field, or programming setting. Writes to a reserved location can result in unpredictable functionality or behavior. • Do not modify the default value of a reserved programming setting, such as the reset value of a reserved register field. • Consider undefined locations in memory to be reserved.
w1c	Write 1 to clear: Refers to a register bitfield that must be written as 1 to be "cleared."

Chapter 2
Introduction

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

2.1 Introduction

The KW36/35 wireless microcontrollers (MCU), which includes the KW36 and KW35 families of devices, are highly integrated single-chip devices that enable Bluetooth Low Energy (BLE) and Generic FSK connectivity for automotive and industrial embedded systems. To meet the stringent requirements of automotive applications, the KW36/35 is fully AEC Q100 Grade 2 Automotive Qualified. The target applications center on wirelessly bridging the embedded world with mobile devices to enhance the human interface experience, share embedded data between devices and the cloud and enable wireless firmware updates. Leading the automotive applications is the Digital Key, where a smartphone can be used by the owner as an alternative to the key FOB for unlocking and personalizing the driving experience. For a car sharing experience, the owner can provide selective, temporary authorization for access to the car allowing the authorized person to unlock, start and operate the car using their mobile device using BLE.

The KW36/35 Wireless MCU integrates an Arm® Cortex-M0+ CPU with up to 512 KB flash and 64 KB SRAM and a 2.4 GHz radio that supports BLE 5.0 and Generic FSK modulations. The BLE radio supports up to 8 simultaneous connections in any master/slave combination. The Medical Body Area Network (MBAN) frequencies from 2.36 to 2.4 GHz are also supported enabling wearable or implantable wireless medical devices.

The KW36 includes an integrated FlexCAN module enabling seamless integration into a cars in-vehicle or industrial CAN communication network. The FlexCAN module can support CAN's flexible data-rate (CAN FD) protocol for increased bandwidth and lower latency required by many automotive applications.

The KW36/35 devices can be used as a "BlackBox" modem in order to add BLE or Generic FSK connectivity to an existing host MCU or MPU (microprocessor), or may be used as a standalone smart wireless sensor with embedded application where no host controller is required.

The RF circuit of the KW36/35 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 through the efficiency of code execution in the Cortex-M0+ CPU core and the multiple low power operating modes of the KW36/35. For power critical applications, an integrated DC-DC converter enables operation from a single coin cell or Li-ion battery with a significant reduction of peak receive and transmit current consumption.

2.2 Block Diagram

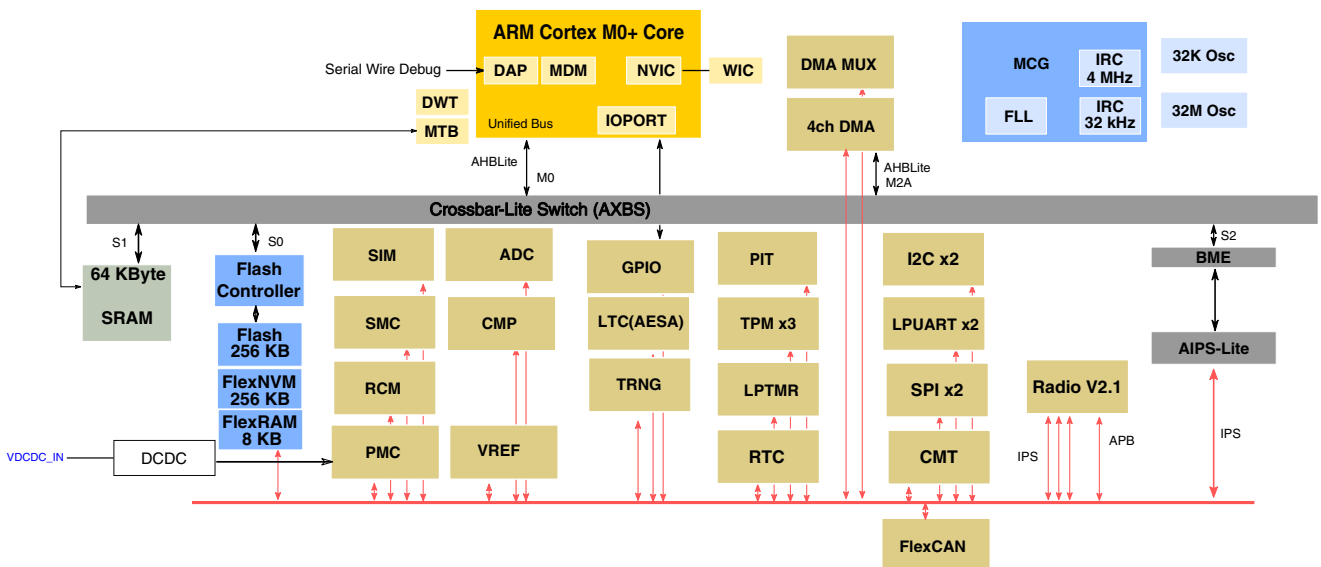


Figure 2-1. KW36 Detailed Block Diagram

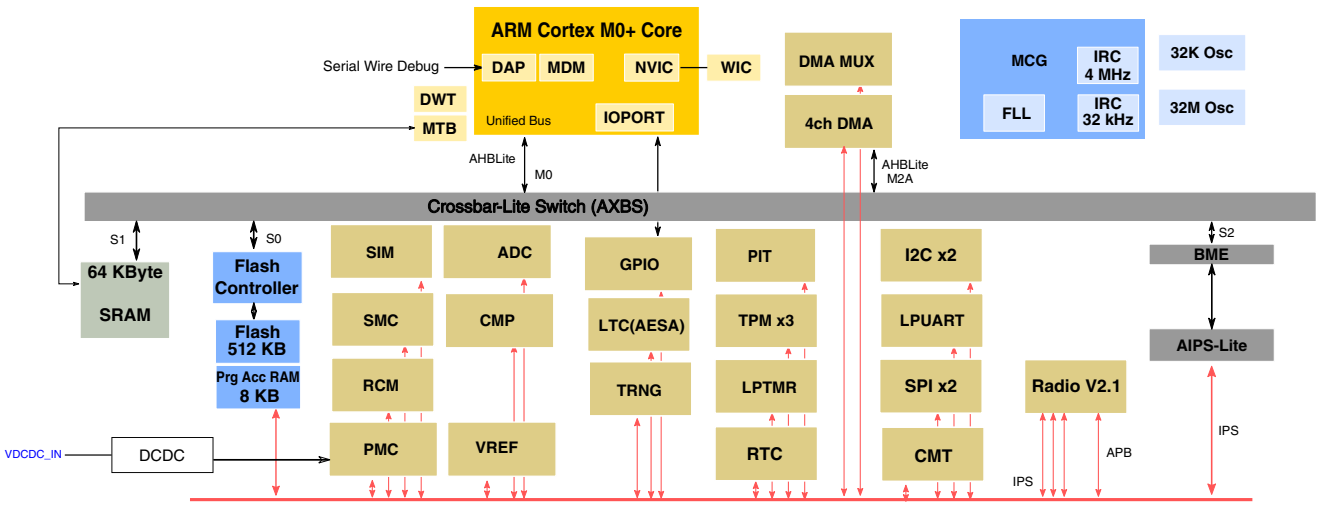


Figure 2-2. KW35 Detailed Block Diagram

2.3 Feature Summary

The following table lists the features integrated on device.

Table 2-1. Feature Summary

Feature	Device
Hardware Characteristics	
Package	48-pin Laminate QFN (7 x 7 mm, 0.5 mm pitch) 40-pin QFN (6 X 6 mm, 0.5 mm pitch) with wettable flanks
Voltage range	DCDC in Buck configurations supports 2.1 V to 3.6 V
	Bypass configuration: 1.71 V to 3.6 V (1.45-3.6 V for RF and OSC supplies)
Temperature range (T_A)	-40° C to +105° C
System	
Central processing unit (CPU)	ARM Cortex-M0+ core (32-bit)
Max. CPU frequency	48 MHz (4 MHz in VLPR mode)
Max. Bus frequency	24 MHz (1 MHz in VLPR mode)
Nested vectored Interrupt controller (NVIC)	32 vectored interrupts 4 programmable interrupt priority levels
Low Power Modes	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,VLLS2, VLLS1, VLLS0)

Table continues on the next page...

Table 2-1. Feature Summary (continued)

Feature	Device
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)
Unique Identification (ID) Numbers	80-bit unique device ID Additional 40-bit unique value for creating MAC address
Memory	
Flash memory	KW35 sub-family: 512 KB P-Flash with ECC and 8 KB Program Acceleration RAM KW36 sub-family: 256 KB P-Flash with ECC, 256 KB FlexNVM, and 8 KB FlexRAM
Random-access memory (RAM)	64 KB
System Register File	32 bytes
Clocks	
Reference crystal oscillator or resonator	Crystal reference oscillator, ability to bypass oscillator with external clock. Supports 26 MHz or 32 MHz crystals
32 kHz External crystal oscillator	Supports 32 kHz 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 4 MHz 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 10 pins (package dependent) with 20 mA high current drive ability Single cycle GPIO control via IOPORT
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 7 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
Voltage Reference(VREF1)	Supply an accurate 1.2 V voltage output
Timers	
16-bit TPM timer (LPTPM x3)	One 4-channel without quadrature decode

Table continues on the next page...

Table 2-1. Feature Summary (continued)

Feature	Device
	Two 2-channel with quadrature decode basic Timer/Pulse-Width Modulator (TPM) function 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 SPI 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 I2C)	Two I2C modules
Low power universal asynchronous receiver/transmitter (2x LPUART)	Two LPUART modules for KW36 and one LPUART module for KW35 Supports LIN break detection 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
FlexCAN	Includes a CAN (FlexCAN0) module that supports Flexible Data-rate (CAN FD) feature (supported in KW36 only)
Radio	
Operating Frequency range	ISM: 2400 - 2483.5 MHz MBAN: 2360 - 2400 MHz
Antenna and RF match support	Support for External, PCB and Ceramic chip antenna
Common Rx/Tx antenna terminals	Use 1 single end pin for Rx and Tx
Extended range options	Capable of supporting an external PA of +30 dB gain Supports +10 dB external LNA
Security Support	
Encryption	AES-128 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

Table continues on the next page...

Table 2-1. Feature Summary (continued)

Feature	Device
	All package pins,including RF pins.
Charge Device Model (CDM) JEDEC STD 2, method C101	+ / -500 V All package pins,including RF pins.

2.4 Features Overview

The following section lists the features of the devices.

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

- Up to 48 MHz core frequency across temperature range of –40 °C to 105 °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 Cortex M0+ 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 configuration supports 2.1 V to 3.6 V
- Bypass Mode 1.7 V to 3.6 V (1.5 V - 3.6 V for the RF and 32 MHz OSC supplies)

On-chip Memory

- 512 KB of flash memory (KW36 has 1x256 KB flash plus 1x256 KB FlexNVM that can execute program code, store data or back up emulated EEPROM data and KW35 has 2x256 KB flash), read/program/erase over full operating voltage and temperature
- 64 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 supporting 32 MHz or 26 MHz crystals
- 32 kHz 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 4 MHz 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-segment (8 KB in KW35 and 4 KB in KW36) basis to prevent firmware contents from being read by third 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.
- VREF — Voltage reference supply accurate voltage output that can be trimmed in 0.5 mV steps

- 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.
- 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-128 Accelerator with DMA support
- TRNG — True Random Number Generator
- LPUART — Serial Communication Interface with DMA support and hardware flow control (RTS\CTS). Supports LIN break detection. KW35 has one LPUART and KW36 has two LPUARTs.
- 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
- FlexCAN — Includes a CAN (FlexCAN0) module that supports Flexible Data-rate (CAN FD) feature (supported in KW36 only)

Radio

- 2.4GHz ISM band (2400-2483.5 MHz) and MBAN 2360-2400 MHz operation
- Supported Standards
  - Bluetooth Low Energy 5
  - Generic FSK modulation capability
- Bluetooth Low Energy Link Layer hardware supporting up to 8 simultaneous connections in any master/slave combination
- 26 MHz or 32 MHz crystal reference oscillator
- Generic FSK Link Layer hardware
- Single RF port shared by both transmit and receive
- Low external component count
- Supports external PA and LNA

NOTE

Refer to the latest version of product datasheet for receiver and transmitter performance and other specifications.

2.5 Orderable Part Numbers

Table 2-2. KW36/35 Part Numbers

Device	Tier	CAN FD	2nd LPUART with LIN	8 KB EEPROM	Package
MKW36A512VFP4	Auto	Y	Y	Y	6X6 mm 40-pin QFN
MKW36Z512VFP4	Industrial	Y	Y	Y	
MKW35A512VFP4	Auto	N	N	N	
MKW36A512VHT4	Auto	Y	Y	Y	7X7 mm 48-pin Laminate QFN
MKW36Z512VHT4	Industrial	Y	Y	Y	
MKW35Z512VHT4	Industrial	N	N	N	

Chapter 3
Signal Multiplexing and Pin Assignment

3.1 Signal Multiplexing and Signal Descriptions

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

3.2 KW36 Signal Multiplexing and Pin Assignments

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

40 QFN	48 LQFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9
—	4	PTA16	DISABLED		PTA16/ LLWU_P4	SPI1_SOUT	LPUART1_ RTS_b		TPM0_CH0				
—	5	PTA17	DISABLED		PTA17/ LLWU_P5	SPI1_SIN	LPUART1_ RX	CAN0_TX	TPM_ CLKIN1				
—	6	PTA18	DISABLED		PTA18/ LLWU_P6	SPI1_SCK	LPUART1_ TX	CAN0_RX	TPM2_CH0				
—	7	PTA19	DISABLED	ADC0_SE5	PTA19/ LLWU_P7	SPI1_PCS0	LPUART1_ CTS_b		TPM2_CH1				
—	24	ADC0_DP0	ADC0_DP0/ CMP0_IN0	ADC0_DP0/ CMP0_IN0									
—	25	ADC0_DM0	ADC0_DM0/ CMP0_IN1	ADC0_DM0/ CMP0_IN1									
—	41	PTC5	DISABLED		PTC5/ LLWU_P13/ RF_NOT_ ALLOWED		LPTMR0_ ALT2	LPUART0_ RTS_b	TPM1_CH1		BSM_CLK		
—	42	PTC6	DISABLED		PTC6/ LLWU_P14/ RF_ RFOSC_EN		I2C1_SCL	LPUART0_ RX	TPM2_CH0		BSM_ FRAME		

40 QFN	48 LQFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9
—	43	PTC7	DISABLED		PTC7/ LLWU_P15	SPI0_PCS2	I2C1_SDA	LPUART0_ TX	TPM2_CH1		BSM_DATA		
1	48	PTC19	DISABLED		PTC19/ LLWU_P3/ RF_ EARLY_ WARNING	SPI0_PCS0	I2C0_SCL	LPUART0_ CTS_b	BSM_CLK			LPUART1_ CTS_b	
2	1	PTA0	SWD_DIO		PTA0	SPI0_PCS1			TPM1_CH0		SWD_DIO		
3	2	PTA1	SWD_CLK		PTA1	SPI1_PCS0			TPM1_CH1		SWD_CLK		
4	3	PTA2	RESET_b		PTA2				TPM0_CH3		RESET_b		
5	8	PSWITCH	PSWITCH	PSWITCH									
6	9	DCDC_CFG	DCDC_CFG	DCDC_CFG									
7	10	VDCDC_IN	VDCDC_IN	VDCDC_IN									
8	11	DCDC_LP	DCDC_LP	DCDC_LP									
9	13	DCDC_GND	DCDC_GND	DCDC_GND									
10	12	DCDC_LN	DCDC_LN	DCDC_LN									
11	14	VDD_ 1P8OUT	VDD_ 1P8OUT	VDD_ 1P8OUT									
12	—	DCDC_LN	DCDC_LN	DCDC_LN									
13	15	VDD_ 1P5OUT_ PMCIN	VDD_ 1P5OUT_ PMCIN	VDD_ 1P5OUT_ PMCIN									
14	16	PTB0	DISABLED		PTB0/ LLWU_P8/ RF_ RFOSC_EN		I2C0_SCL	CMP0_OUT	TPM0_CH1		CLKOUT	CAN0_TX	
15	17	PTB1	DISABLED	ADC0_SE1/ CMP0_IN5	PTB1/ RF_ PRIORITY	DTM_RX	I2C0_SDA	LPTMR0_ ALT1	TPM0_CH2		CMT_IRO	CAN0_RX	
16	18	PTB2	DISABLED	ADC0_SE3/ CMP0_IN3	PTB2/ RF_NOT_ ALLOWED		DTM_TX		TPM1_CH0				
17	19	PTB3	DISABLED	ADC0_SE2/ CMP0_IN4	PTB3/ ERCLK32K	LPUART1_ RTS_b		CLKOUT	TPM1_CH1		RTC_ CLKOUT		
18	20	VDD_0	VDD_0	VDD_0									
19	21	PTB16	EXTAL32K	EXTAL32K	PTB16	LPUART1_ RX	I2C1_SCL		TPM2_CH0				
20	22	PTB17	XTAL32K	XTAL32K	PTB17	LPUART1_ TX	I2C1_SDA		TPM2_CH1		BSM_CLK		
21	23	PTB18	NMI_b	ADC0_SE4/ CMP0_IN2	PTB18	LPUART1_ CTS_b	I2C1_SCL	TPM_ CLKIN0	TPM0_CH0		NMI_b		
22	26	VREFL/ VSSA	VREFL/ VSSA	VREFL/ VSSA									
23	27	VREFH/ VREF_OUT	VREFH/ VREF_OUT	VREFH/ VREF_OUT									
24	28	VDDA	VDDA	VDDA									

40 QFN	48 LQFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9
25	29	XTAL_OUT	XTAL_OUT	XTAL_OUT									
26	30	EXTAL	EXTAL	EXTAL									
27	31	XTAL	XTAL	XTAL									
28	32	VDD_RF3	VDD_RF3	VDD_RF3									
29	33	ANT	ANT	ANT									
30	34	GANT	GANT	GANT									
31	35	VDD_RF2	VDD_RF2	VDD_RF2									
32	36	VDD_RF1	VDD_RF1	VDD_RF1									
33	37	PTC1	DISABLED		PTC1/ RF_ EARLY_ WARNING		I2C0_SDA	LPUART0_ RTS_b	TPM0_CH2			SPI1_SCK	BSM_CLK
34	38	PTC2	DISABLED		PTC2/ LLWU_P10	TX_ SWITCH	I2C1_SCL	LPUART0_ RX	CMT_IRO		DTM_RX	SPI1_SOUT	BSM_ FRAME
35	39	PTC3	DISABLED		PTC3/ LLWU_P11	RX_ SWITCH	I2C1_SDA	LPUART0_ TX	TPM0_CH1		DTM_TX	SPI1_SIN	CAN0_TX
36	40	PTC4	DISABLED		PTC4/ LLWU_P12/ BLE_RF_ ACTIVE		EXTRG_IN	LPUART0_ CTS_b	TPM1_CH0		BSM_DATA	SPI1_PCS0	CAN0_RX
37	44	VDD_1	VDD_1	VDD_1									
38	45	PTC16	DISABLED		PTC16/ LLWU_P0/ RF_ STATUS	SPI0_SCK	I2C0_SDA	LPUART0_ RTS_b	TPM0_CH3			LPUART1_ RTS_b	
39	46	PTC17	DISABLED		PTC17/ LLWU_P1/ RF_EXT_ OSC_EN	SPI0_SOUT	I2C1_SCL	LPUART0_ RX	BSM_ FRAME		DTM_RX	LPUART1_ RX	
40	47	PTC18	DISABLED		PTC18/ LLWU_P2	SPI0_SIN	I2C1_SDA	LPUART0_ TX	BSM_DATA		DTM_TX	LPUART1_ TX	
41	49-64	Ground	NA										

3.3 KW36 Pinouts

KW36 device pinouts are shown in the figures below.

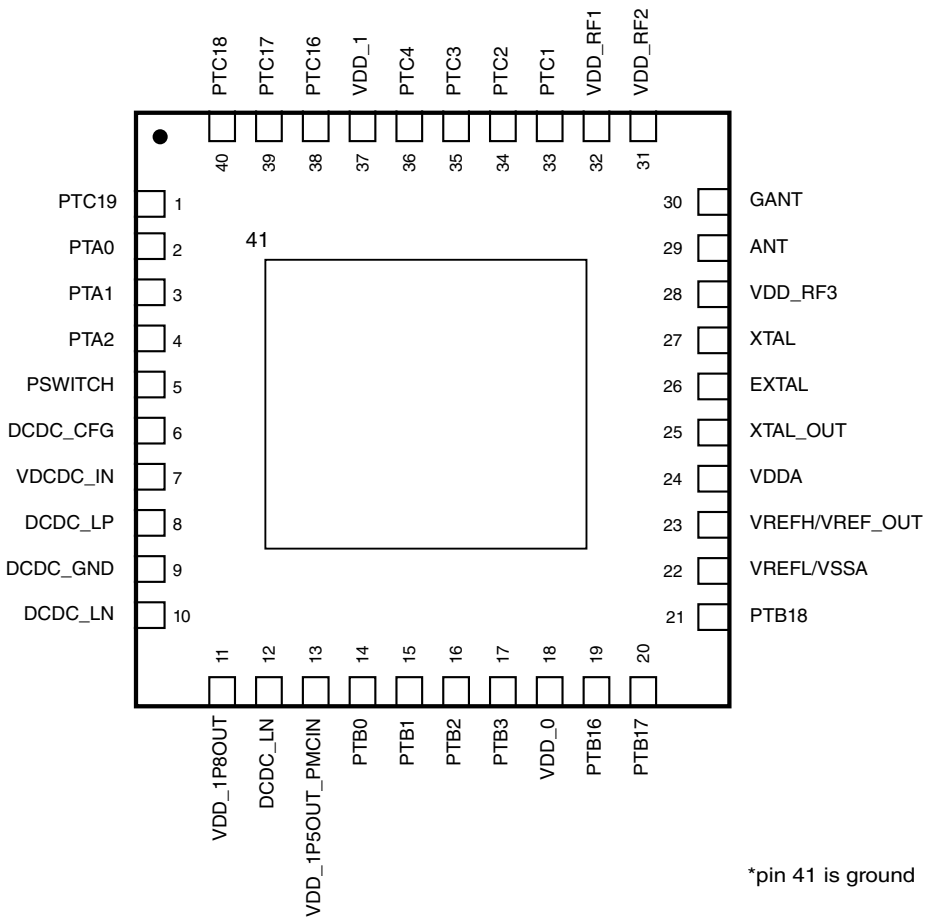


Figure 3-1. 40-pin QFN pinout diagram

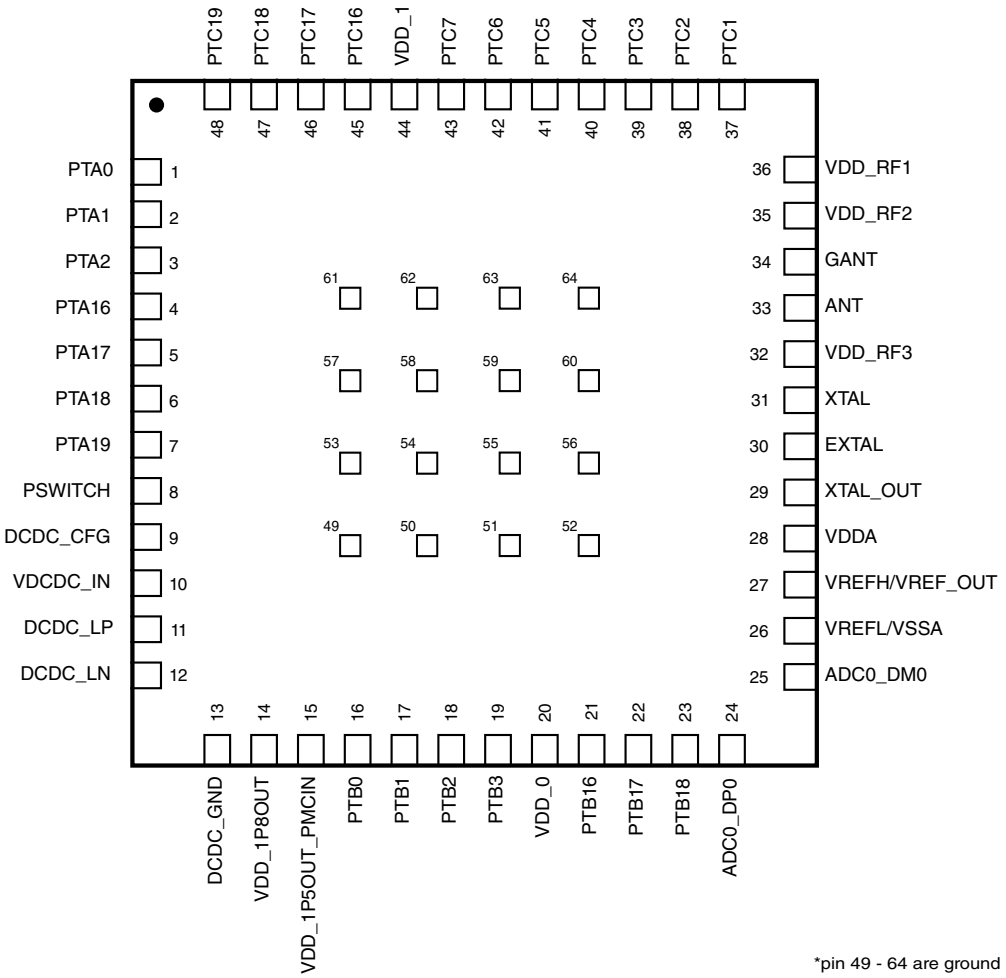


Figure 3-2. 48-pin LQFN pinout diagram

3.4 KW35 Signal Multiplexing and Pin Assignments

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

40 QFN	48 LQFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9
—	4	PTA16	DISABLED		PTA16/ LLWU_P4	SPI1_SOUT			TPM0_CH0				
—	5	PTA17	DISABLED		PTA17/ LLWU_P5	SPI1_SIN			TPM_ CLKIN1				

40 QFN	48 LQFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9
—	6	PTA18	DISABLED		PTA18/ LLWU_P6	SPI1_SCK			TPM2_CH0				
—	7	PTA19	DISABLED	ADC0_SE5	PTA19/ LLWU_P7	SPI1_PCS0			TPM2_CH1				
—	24	ADC0_DP0	ADC0_DP0/ CMP0_IN0	ADC0_DP0/ CMP0_IN0									
—	25	ADC0_DM0	ADC0_DM0/ CMP0_IN1	ADC0_DM0/ CMP0_IN1									
—	41	PTC5	DISABLED		PTC5/ LLWU_P13/ RF_NOT_ ALLOWED		LPTMR0_ ALT2	LPUART0_ RTS_b	TPM1_CH1		BSM_CLK		
—	42	PTC6	DISABLED		PTC6/ LLWU_P14/ RF_ RFOSC_EN		I2C1_SCL	LPUART0_ RX	TPM2_CH0		BSM_ FRAME		
—	43	PTC7	DISABLED		PTC7/ LLWU_P15	SPI0_PCS2	I2C1_SDA	LPUART0_ TX	TPM2_CH1		BSM_DATA		
1	48	PTC19	DISABLED		PTC19/ LLWU_P3/ RF_ EARLY_ WARNING	SPI0_PCS0	I2C0_SCL	LPUART0_ CTS_b	BSM_CLK				
2	1	PTA0	SWD_DIO		PTA0	SPI0_PCS1			TPM1_CH0		SWD_DIO		
3	2	PTA1	SWD_CLK		PTA1	SPI1_PCS0			TPM1_CH1		SWD_CLK		
4	3	PTA2	RESET_b		PTA2				TPM0_CH3		RESET_b		
5	8	PSWITCH	PSWITCH	PSWITCH									
6	9	DCDC_CFG	DCDC_CFG	DCDC_CFG									
7	10	VDCDC_IN	VDCDC_IN	VDCDC_IN									
8	11	DCDC_LP	DCDC_LP	DCDC_LP									
9	13	DCDC_GND	DCDC_GND	DCDC_GND									
10	12	DCDC_LN	DCDC_LN	DCDC_LN									
11	14	VDD_ 1P8OUT	VDD_ 1P8OUT	VDD_ 1P8OUT									
12	—	DCDC_LN	DCDC_LN	DCDC_LN									
13	15	VDD_ 1P5OUT_ PMCIN	VDD_ 1P5OUT_ PMCIN	VDD_ 1P5OUT_ PMCIN									
14	16	PTB0	DISABLED		PTB0/ LLWU_P8/ RF_ RFOSC_EN		I2C0_SCL	CMP0_OUT	TPM0_CH1		CLKOUT		
15	17	PTB1	DISABLED	ADC0_SE1/ CMP0_IN5	PTB1/ RF_ PRIORITY	DTM_RX	I2C0_SDA	LPTMR0_ ALT1	TPM0_CH2		CMT_IRO		
16	18	PTB2	DISABLED	ADC0_SE3/ CMP0_IN3	PTB2/ RF_NOT_ ALLOWED		DTM_TX		TPM1_CH0				

40 QFN	48 LQFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9
17	19	PTB3	DISABLED	ADC0_SE2/ CMP0_IN4	PTB3/ ERCLK32K			CLKOUT	TPM1_CH1		RTC_ CLKOUT		
18	20	VDD_0	VDD_0	VDD_0									
19	21	PTB16	EXTAL32K	EXTAL32K	PTB16		I2C1_SCL		TPM2_CH0				
20	22	PTB17	XTAL32K	XTAL32K	PTB17		I2C1_SDA		TPM2_CH1		BSM_CLK		
21	23	PTB18	NMI_b	ADC0_SE4/ CMP0_IN2	PTB18		I2C1_SCL	TPM_ CLKIN0	TPM0_CH0		NMI_b		
22	26	VREFL/ VSSA	VREFL/ VSSA	VREFL/ VSSA									
23	27	VREFH/ VREF_OUT	VREFH/ VREF_OUT	VREFH/ VREF_OUT									
24	28	VDDA	VDDA	VDDA									
25	29	XTAL_OUT	XTAL_OUT	XTAL_OUT									
26	30	EXTAL	EXTAL	EXTAL									
27	31	XTAL	XTAL	XTAL									
28	32	VDD_RF3	VDD_RF3	VDD_RF3									
29	33	ANT	ANT	ANT									
30	34	GANT	GANT	GANT									
31	35	VDD_RF2	VDD_RF2	VDD_RF2									
32	36	VDD_RF1	VDD_RF1	VDD_RF1									
33	37	PTC1	DISABLED		PTC1/ RF_ EARLY_ WARNING		I2C0_SDA	LPUART0_ RTS_b	TPM0_CH2			SPI1_SCK	BSM_CLK
34	38	PTC2	DISABLED		PTC2/ LLWU_P10	TX_ SWITCH	I2C1_SCL	LPUART0_ RX	CMT_IRO		DTM_RX	SPI1_SOUT	BSM_ FRAME
35	39	PTC3	DISABLED		PTC3/ LLWU_P11	RX_ SWITCH	I2C1_SDA	LPUART0_ TX	TPM0_CH1		DTM_TX	SPI1_SIN	
36	40	PTC4	DISABLED		PTC4/ LLWU_P12/ BLE_RF_ ACTIVE		EXTRG_IN	LPUART0_ CTS_b	TPM1_CH0		BSM_DATA	SPI1_PCS0	
37	44	VDD_1	VDD_1	VDD_1									
38	45	PTC16	DISABLED		PTC16/ LLWU_P0/ RF_ STATUS	SPI0_SCK	I2C0_SDA	LPUART0_ RTS_b	TPM0_CH3				
39	46	PTC17	DISABLED		PTC17/ LLWU_P1/ RF_EXT_ OSC_EN	SPI0_SOUT	I2C1_SCL	LPUART0_ RX	BSM_ FRAME		DTM_RX		
40	47	PTC18	DISABLED		PTC18/ LLWU_P2	SPI0_SIN	I2C1_SDA	LPUART0_ TX	BSM_DATA		DTM_TX		
41	49-64	Ground	NA										

3.5 KW35 Pinouts

KW35 device pinouts are shown in the figures below.

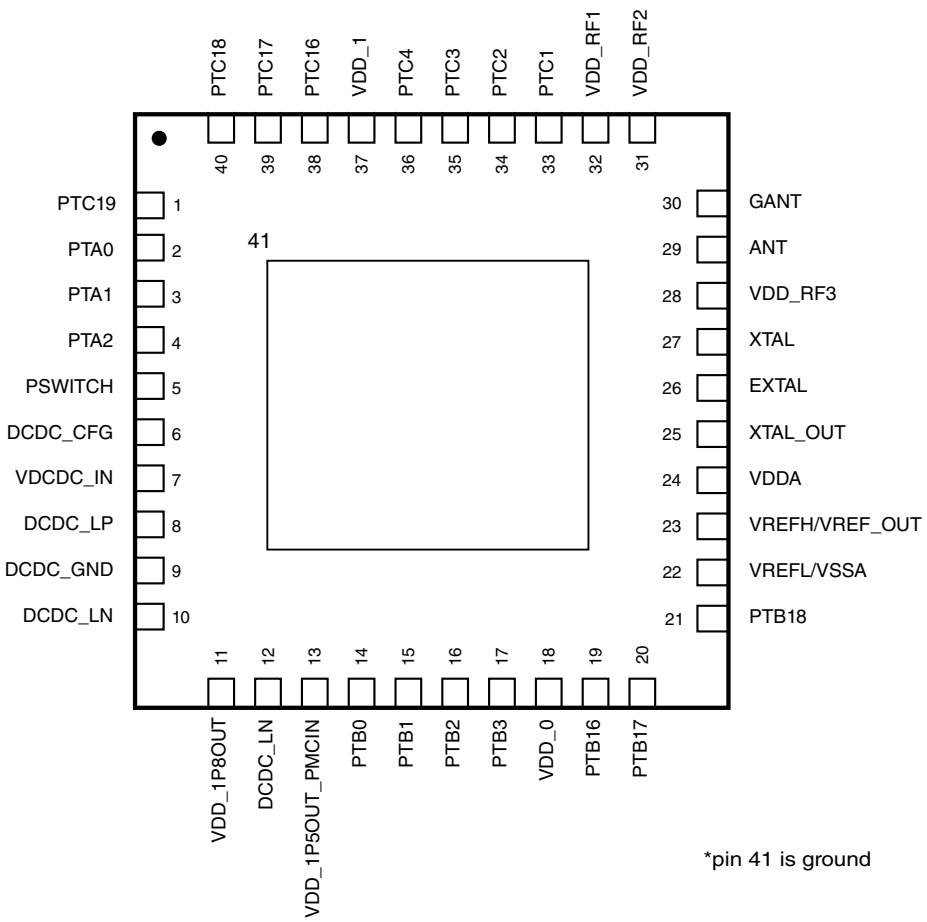


Figure 3-3. 40-pin QFN pinout diagram

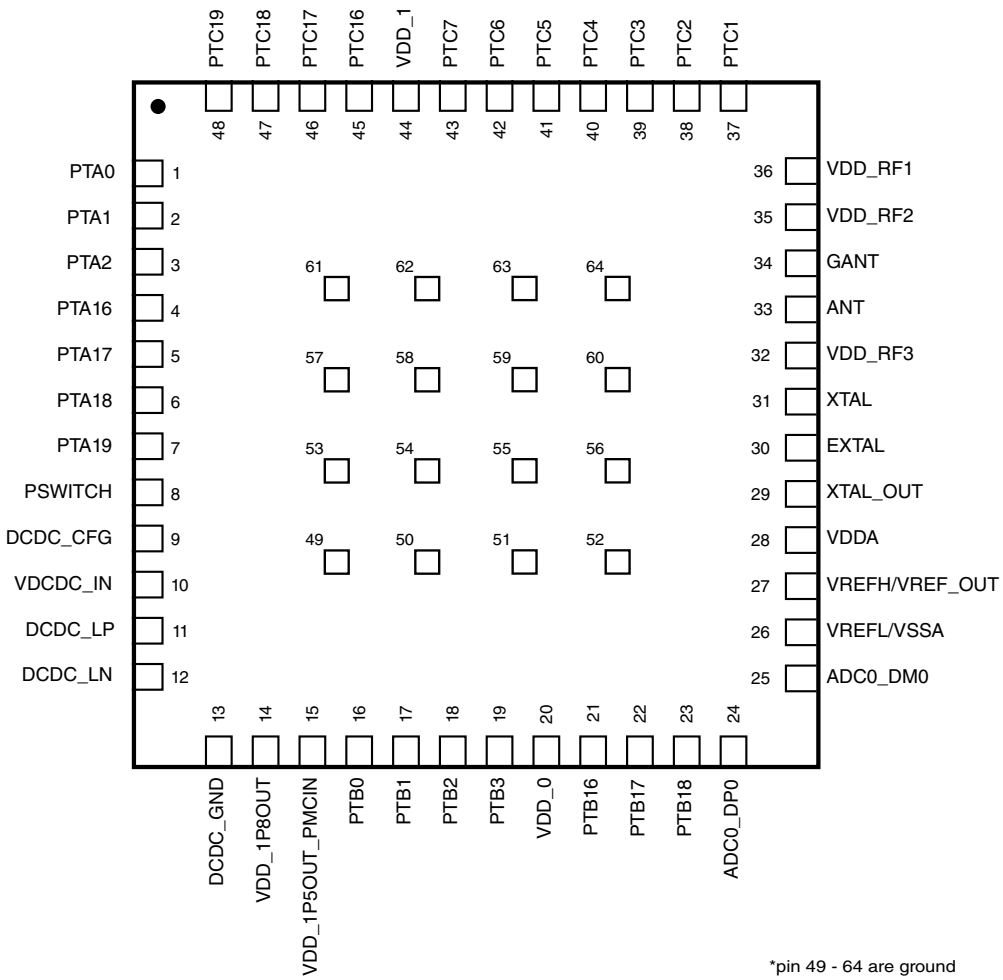


Figure 3-4. 48-pin LQFN pinout diagram

3.6 KW36/35 Signal Descriptions

Table 3-1. KW36/35 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_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
ADC0_SE5	ADC Channel 0 Single-ended Input 5	A

Table continues on the next page...

Table 3-1. KW36/35 Signal Descriptions (continued)

Signal Name	Description	Type
ANT	Antenna	A
BLE_RF_ACTIVE	Signal to indicate future BLE activity. Refer BLE Link Layer for more details.	D
BSM_CLK	Bit Streaming Mode Clock	D
BSM_DATA	Bit Streaming Mode Data	D
BSM_FRAME	Bit Streaming Mode Frame	D
CLKOUT	Internal Clocks Monitoring	D
CMP0_IN0	Comparator0 Input 0	A
CMP0_IN1	Comparator0 Input 0	A
CMP0_IN2	Comparator0 Input 0	A
CMP0_IN3	Comparator0 Input 0	A
CMP0_IN4	Comparator0 Input 0	A
CMP0_IN5	Comparator0 Input 0	A
CMP0_OUT	Comparator0 Output	D
CMT_IRO	Carrier Modulator Transmitter Infrared Output	D
DCDC_CFG	DCDC Switch Mode Select	D
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	26 MHz or 32 MHz Crystal Input	A
EXTAL32K	32kHz Crystal Input	A
EXTRG_IN	TPM or ADC External Trigger Input	D
GANT	Antenna ground	A
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

Table continues on the next page...

Table 3-1. KW36/35 Signal Descriptions (continued)

Signal Name	Description	Type
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
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
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_EARLY_WARNING	BLE LL generated signal which can be used to wake an external sensor to make a measurement before a BLE event.	D
RF_EXT_OSC_EN	Internal request to turn on an external oscillator for use by the internal Radio. The request can also be from the SoC if it is using the RF oscillator as its clock.	D
RF_PRIORITY	An output that notifies to the external WiFi device that the Radio event is a high priority and needs access to the 2.4 GHz antenna.	D
RF_STATUS	An output which indicates when the Radio is in an RX or a TX event; software can also control this signal directly.	D
RTC_CLKOUT	RTC Clock Out	D
RF_NOT_ALLOWED	Input signal that allows the external chip (eg. WiFi Chip) to signal the 2.4 GHz radio when BTLE and Generic FSK LL radio operations are not allowed.	D
RF_RFOSC_EN	External request to turn on the Radio's internal RF oscillator. The oscillator will be present on the XTAL_OUT pin if the RF oscillator XTAL_OUT buffer is enabled by writing the Radio XTAL_OUT_BUF_EN bit in the RSIM ANA_TEST register.	D
RX_SWITCH	Supports Front End Module (FEM)	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

Table continues on the next page...

Table 3-1. KW36/35 Signal Descriptions (continued)

Signal Name	Description	Type
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
TX_SWITCH	Supports Front End Module (FEM)	D
LPUART0_CTS_b	LPUART0 Clear To Send	D
LPUART0_RTS_b	LPUART0 Request To Send	D
LPUART0_RX	LPUART0 Receive	D
LPUART0_TX	LPUART0 Transmit	D
LPUART1_CTS_b	LPUART1 Clear To Send	D
LPUART1_RTS_b	LPUART1 Request To Send	D
LPUART1_RX	LPUART1 Receive	D
LPUART1_TX	LPUART1 Transmit	D
VDCDC_IN	DCDC Switch Main Supply	P
VDD_0	Power Supply 0	P
VDD_1	Power Supply 1	P
VDD_1P5OUT_PMCIN	DCDC Pulsed Output 1.5 V Regulated Output or PMC Input when DCDC in bypass mode	P
VDD_1P8OUT	DCDC Pulsed 1.8 V Regulated Output	P
VDDA	Power Supply - Analog	P
VREFH	ADC reference voltage	P
VSSA	ADC ground	G
XTAL	26 MHz or 32 MHz Crystal Input	A
XTAL32K	32 kHz Crystal Input	A

Legend

- A - Analog

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

3.7 Module Signal Description Tables

The following sections correlate the chip-level signal name with the signal name used in the module's chapter. They also briefly describe the signal function and direction.

3.7.1 Core Modules

This section contains tables describing the core module signal descriptions.

Table 3-2. SWD Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
SWD_DIO	SWD_DIO	Serial Wire Debug Data Input/ Output[1]	I/O
SWD_CLK	SWD_CLK	Serial Wire Clock[2]	I

1. Pulled up internally by default
2. Pulled down internally by default

3.7.2 Radio Modules

This section contains tables describing the radio signals.

Table 3-3. Radio Module Signal Descriptions

Module Signal Name	Pin Direction	Pin Name	Pin Description
ANT	O	ANT	Antenna
BLE_RF_ACTIVE	O	BLE_RF_ACTIVE	An output which is asserted prior to any Radio event and remains asserted for the duration of the event.
BSM_CLK	O	BSM_CLK	Bit Streaming Mode (BSM) clock signal. 1 MHz bit rate clock. BSM_DATA and BSM_FRAME are synchronized to BSM_CLK. External device should capture BSM_FRAME and BSM_DATA on rising edge of BSM_CLK.
BSM_DATA	O	BSM_DATA	Serial BLE packet bit stream, LSB-first. Valid on rising edge of BSM_CLK.
BSM_FRAME	O	BSM_FRAME	Framing signal to indicate the start of reception. Active high.

Table continues on the next page...

Table 3-3. Radio Module Signal Descriptions (continued)

Module Signal Name	Pin Direction	Pin Name	Pin Description
DTM_RX	I	DTM_RX	Direct Test Mode Receive
DTM_TX	O	DTM_TX	Direct Test Mode Transmit
GANT	I	GANT	Antenna ground
RF_STATUS	O	RF_STATUS	An output which indicates when the Radio is in an RX or TX event; software can also control this signal directly.
RF_PRIORITY	O	RF_PRIORITY	An output which indicates to the external WiFi device that the Radio event is a high priority and it needs access to the 2.4GHz antenna.
RF_EARLY_WARNING	O	RF_EARLY_WARNING	BLE LL generated signal which can be used to wake an external sensor to make a measurement before a BLE event.
RF_NOT_ALLOWED	I	RF_NOT_ALLOWED	External signal which causes the internal Radio to cease radio activity.
RF_TX_CONF	I	RF_TX_CONF	Signal from an external Radio which indicates the availability of the 2.4GHz antenna to the internal Radio. **NOTE:** This is a GPIO, not a dedicated PIN.
RX_SWITCH	O	TX_SWITCH	Front End Module receive mode signal.
TX_SWITCH	O	TX_SWITCH	Front End Module transmit mode signal.

Table 3-4. Radio Module Miscellaneous Pin Descriptions

Pin Name	Pad Direction	Pin Name	Pin Description
RF_INT_OSC_EN	I	RF_RFOSC_EN	External request to turn on the Radio's internal RF oscillator. The oscillator will be present on the XTAL_OUT pin if the RF oscillator XTAL_OUT buffer is enabled by writing the Radio XTAL_OUT_BUF_EN bit in the RSIM ANA_TEST register.
RF_EXT_OSC_EN	O	RF_EXT_OSC_EN	Internal request to turn on an External oscillator for use by the internal Radio. The request can also be from the SoC if it is using the RF oscillator as its clock.

3.7.3 System Modules

This section contains tables describing the system signals.

Table 3-5. System Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
NMI_b	—	Non-maskable interrupt	I
RESET_b	—	Reset bidirectional signal	I/O

Table continues on the next page...

Table 3-5. System Module Signal Descriptions (continued)

SoC Signal Name	Module Signal Name	Description	I/O
VDD_[1:0]	VDD	Power supply	I
Ground	VSS	Ground	I
VDD_RF[3:1]	VDD_RF	Radio power supply	I
VDCDC_IN	VDCDC_IN	VDCDC_IN	I
VDD_1P8OUT	VDD_1P8	DCDC 1.8 V Regulated Output / Input in bypass	I/O
VDD_1P5OUT_PMCIN	VDD_1P5/VDD_PMC	DCDC 1.5 V Regulated Output / PMC Input in bypass	I/O
PSWITCH	PSWITCH	DCDC enable switch	I
DCDC_CFG	DCDC_CFG	DCDC switch mode select	I
DCDC_LP	DCDC_LP	DCDC inductor input positive	I/O
DCDC_LN	DCDC_LN	DCDC inductor input negative	I/O
DCDC_GND	DCDC_GND	DCDC ground	I

Table 3-6. LLWU Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
LLWU_P[15:0]	LLWU_P[15:0]	Wakeup inputs	I

3.7.4 Clock Modules

This section contains tables for Clock signal descriptions.

Table 3-7. Clock Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
EXTAL	EXTAL	26 MHz/32 MHz External clock/Oscillator input	I
XTAL	XTAL	26 MHz/32 MHz Oscillator input	I
XTAL_OUT	XTAL_OUT	26 MHz/32 MHz Clock output	O
XTAL_OUT_EN	XTAL_OUT_ENABLE	26 MHz/32 MHz Clock output enable for XTAL_OUT	I
EXTAL32K	EXTAL32K	32 kHz External clock/ Oscillator input	I
XTAL32K	XTAL32K	32 kHz Oscillator input	I
CLKOUT	CLKOUT	Internal clocks monitor	O

3.7.5 Analog Modules

This section contains tables for Analog signal descriptions.

Table 3-8. ADC0 Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
ADC0_DM0	DADM0	ADC Channel 0 Differential Input Negative	I
ADC0_DP0	DADP0	ADC Channel 0 Differential Input Positive	I
ADC0_SE[5:1]	AD[5:1]	ADC Channel 0 Single-ended Input n	I
VREFH	V_{REFSH}	Voltage Reference Select High	I
VDDA	V_{DDA}	Analog Power Supply	I
VSSA	V_{SSA}	Analog Ground	I

Table 3-9. CMP0 Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
CMP0_IN[5:0]	IN[5:0]	Analog voltage inputs	I
CMP0_OUT	CMP0	Comparator output	O

Table 3-10. VREF Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
VREF_OUT	VREF_OUT	Internally generated voltage reference output	O

3.7.6 Timer Modules

This section contains tables describing timer module signals.

Table 3-11. TPM0 Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
TPM_CLKIN[1:0]	TPM_EXTCLK	External clock	I
TPM0_CH[3:0]	TPM_CH[3:0]	TPM channel	I/O

Table 3-12. TPM1 Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
TPM_CLKIN[1:0]	TPM_EXTCLK	External clock	I
TPM1_CH[1:0]	TPM_CH[1:0]	TPM channel	I/O

Table 3-13. TPM2 Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
TPM_CLKIN[1:0]	TPM_EXTCLK	External clock	I
TPM2_CH[1:0]	TPM_CH[1:0]	TPM channel	I/O

Table 3-14. LPTMR0 Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
LPTMR0_ALT[2:1]	LPTMR0_ALT[2:1]	Pulse counter input pin	I

Table 3-15. RTC Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
RTC_CLKOUT	RTC_CLKOUT	1 Hz square-wave output	O

3.7.7 Communication Interfaces

This section contains tables for the signal descriptions for the communication modules.

Table 3-16. SPI0 Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
SPI0_PCS0	PCS0/SS	Chip Select/Slave Select	I/O
SPI0_PCS[2:1]	PCS[2:1]	Chip Select	O
SPI0_SCK	SCK	Serial Clock	I/O
SPI0_SIN	SIN	Data In	I
SPI0_SOUT	SOUT	Data Out	O

Table 3-17. SPI1 Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
SPI1_PCS0	SPI1_PCS0	Chip Select/Slave Select	I/O
SPI1_SCK	SCK	Serial Clock	I/O
SPI1_SIN	SIN	Data In	I
SPI1_SOUT	SOUT	Data Out	O

Table 3-18. I2C0 Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
I2C0_SCL	SCL	I2C serial clock line	I/O
I2C0_SDA	SDA	I2C serial data line	I/O

Table 3-19. I2C1 Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
I2C1_SCL	SCL	I2C serial clock line	I/O
I2C1_SDA	SDA	I2C serial data line	I/O

Table 3-20. CAN0 Signal Descriptions (KW36 only)

SoC Signal Name	Module Signal Name	Description	I/O
CAN0_RX	CAN RX	CAN Receive Pin	I
CAN0_TX	CAN TX	CAN Transmit Pin	O

Table 3-21. LPUART0 Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
LPUART0_CTS_b	LPUART CTS	Clear To Send	I
LPUART0_RTS_b	LPUART RTS	Request To Send	O
LPUART0_RX	LPUART RxD	Receive Data	I
LPUART0_TX	LPUART TxD	Transmit Data[1]	I/O

1. 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

Table 3-22. LPUART1 Module Signal Descriptions (KW36 only)

SoC Signal Name	Module Signal Name	Description	I/O
LPUART1_CTS_b	LPUART CTS	Clear To Send	I
LPUART1_RTS_b	LPUART RTS	Request To Send	O
LPUART1_RX	LPUART RxD	Receive Data	I
LPUART1_TX	LPUART TxD	Transmit Data[1]	I/O

1. 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

3.7.8 Human-Machine Interfaces(HMI)

This section contains tables describing the HMI signals.

Table 3-23. GPIO Module Signal Descriptions

SoC Signal Name	Module Signal Name	Description	I/O
PTA[19:16][2:0]	PORTA19-16, 2-0	General Purpose Input/Output	I/O
PTB[18:16][3:0]	PORTB18-16, 3-0	General Purpose Input/Output	I/O
PTC[19:16][7:1]	PORTC19-16, 7-1	General Purpose Input/Output	I/O

Chapter 4
Chip Configuration

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

4.1 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

4.2 Module to module interconnects

4.2.1 Module to Module Interconnects

The below table captures the Module to module interconnections for this device.

Table 4-1. Module to Module Interconnects

Peripheral	Signal	to Peripheral	Use Case	Control	Comment
TPM1	CH0F, CH1F	ADC (Trigger)	ADC Triggering (A AND B)	SOPT7[ADC0ALTTRGEN] = 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 trigger must be set to >10µs after the first trigger.

Table continues on the next page...

Table 4-1. Module to Module Interconnects (continued)

Peripheral	Signal	to Peripheral	Use Case	Control	Comment
TPMx	TOF	ADC (Trigger)	ADC Triggering (A OR B)	SOPT7[ADC0TRGSEL], SOPT7[ADC0PRETRGSEL] to select A or B	—
LPTMR	Hardware trigger	ADC (Trigger)	ADC Triggering (A OR B)	SOPT7[ADC0TRGSEL], SOPT7[ADC0PRETRGSEL] to select A or B	—
PIT CHx	TIF0, TIF1	ADC (Trigger)	ADC Triggering (A OR B)	SOPT7[ADC0TRGSEL], SOPT7[ADC0PRETRGSEL] to select A or B	—
RTC	ALARM or SECONDS	ADC (Trigger)	ADC Triggering (A OR B)	SOPT7[ADC0TRGSEL], SOPT7[ADC0PRETRGSEL] to select A or B	—
EXTRG_IN	EXTRG_IN	ADC (Trigger)	ADC Triggering (A OR B)	SOPT7[ADC0TRGSEL], SOPT7[ADC0PRETRGSEL] to select A or B	—
CMP0	CMP0_OUT	ADC (Trigger)	ADC Triggering (A OR B)	SOPT7[ADC0TRGSEL], SOPT7[ADC0PRETRGSEL] to select A or B	—
Radio TSM	sar_adc_trig	ADC (Trigger)	ADC Triggering (A OR B).	SOPT7[ADC0TRGSEL], SOPT7[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	LPTMR_ALT0	Count CMP events	LPTMR_CSR[TPS]	
CMP0	CMP0_OUT	TPM1 CH0	Input capture	SOPT4[TPM1CH0SRC]	
CMP0	CMP0_OUT	TPM2 CH0	Input capture	SOPT4[TPM2CH0SRC]	
CMP0	CMP0_OUT	LPUART0_RX	IR interface	SOPT5[LPUART0RXSRC]	
CMP0	CMP0_OUT	LPUART1_RX	IR interface	SOPT5[LPUART1RXSRC]	
LPTMR	Hardware trigger	CMP0	Low power triggering of the comparator	CMP_CR1[TRIGM]	

Table continues on the next page...

Table 4-1. Module to Module Interconnects (continued)

Peripheral	Signal	to Peripheral	Use Case	Control	Comment
LPTMR	Hardware trigger	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL]	—
TPMx	TOF	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL]	—
PIT CHx	TIF0, TIF1	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL]	If PIT is triggering the TPM, the TPM clock must be faster than Bus clock.
RTC	ALARM or SECONDS	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL]	—
EXTRG_IN	EXTRG_IN	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL]	—
CMP0	CMP0_OUT	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL]	
Radio TSM	sar_adc_trig[1]	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL]	
TPM1	Timebase	TPMx	TPM Global timebase input	TPMx_CONF[GTBEEN]	—
LPUART0	LPUART0_TX	Modulated by TPM1 CH0	LPUART modulation	SOPT5[LPUART0TXSRC]	—
LPUART0	LPUART0_TX	Modulated by TPM2 CH0	LPUART modulation	SOPT5[LPUART0TXSRC]	—
LPUART1	LPUART1_TX	Modulated by TPM1 CH0	LPUART modulation	SOPT5[LPUART1TXSRC]	—
LPUART1	LPUART1_TX	Modulated by TPM2 CH0	LPUART modulation	SOPT5[LPUART1TXSRC]	—
PIT	TIF0	DMA CH0	DMA HW Trigger	DMA MUX register option	
PIT	TIF1	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.

4.3 Core modules

4.3.1 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**.

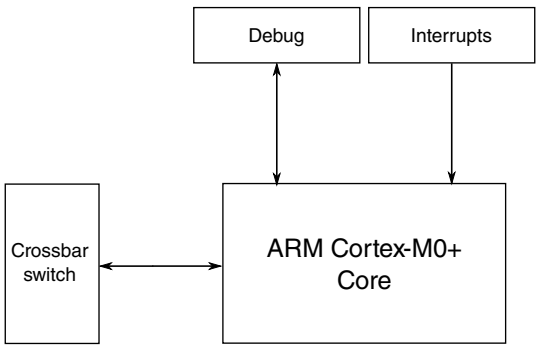


Figure 4-1. Core configuration

Table 4-2. 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

4.3.1.1 ARM Cortex M0+ core

The ARM Cortex M0+ parameter settings are as follows:

Table 4-3. 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	—

Table continues on the next page...

Table 4-3. ARM Cortex-M0+ parameter settings (continued)

Parameter	Value	Description
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 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**.

4.3.1.2 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.

4.3.1.3 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.

4.3.1.4 Debug Facilities

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

4.3.1.5 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

4.3.2 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**.

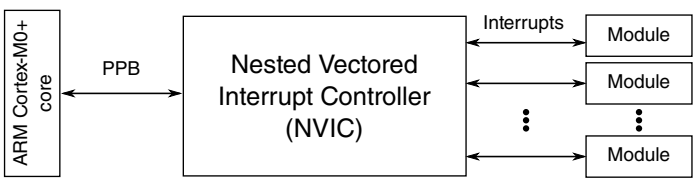


Figure 4-2. NVIC configuration

Table 4-4. 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
Clocking		Clock distribution
Power management		Power management
Private Peripheral Bus (PPB)	ARM Cortex-M0+ core	ARM Cortex-M0+ core

4.3.2.1 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																																

4.3.2.2 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. 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.

4.3.2.3 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 4-6. Interrupt vector assignments

Address	Vector	IRQ[1]	Source module	Source description
Non-core Vectors				
On-platform Vectors				
0x0000_0040	16	0	DMA	DMA channel 0 transfer complete and error
0x0000_0044	17	1	DMA	DMA channel 1 transfer complete and error
0x0000_0048	18	2	DMA	DMA channel 2 transfer complete and error

Table continues on the next page...

Table 4-6. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	Source module	Source description
0x0000_004C	19	3	DMA	DMA channel 3 transfer complete and error
0x0000_0050	20	4	—	Reserved for future MCM
Off-platform Vectors				
0x0000_0054	21	5	FTFE	Command complete and read collision
0x0000_0058	22	6	PMC and DCDC	PMC: Low-voltage detect, low-voltage warning DCDC: PSWITCH interrupt
0x0000_005C	23	7	LLWU	Low Leakage Wakeup
0x0000_0060	24	8	I2C0	
0x0000_0064	25	9	I2C1	
0x0000_0068	26	10	SPI0	Single interrupt vector for all sources
0x0000_006C	27	11	• — (for KW35) • FlexCAN0 (for KW36)	FlexCAN0 ORed Error, Bus off, Transmit/Receive Warning, Wake up Interrupts (for KW36 only)
0x0000_0070	28	12	UART0 or UART1 (LPUART)	Single interrupt vector for for UART0 and UART1 sources
0x0000_0074	29	13	TRNG	
0x0000_0078	30	14	CMT	Status and error
0x0000_007C	31	15	ADC0	
0x0000_0080	32	16	CMP0	
0x0000_0084	33	17	TPM0	
0x0000_0088	34	18	TPM1	
0x0000_008C	35	19	TPM2	
0x0000_0090	36	20	RTC	Alarm interrupt
0x0000_0094	37	21	RTC	Seconds interrupt
0x0000_0098	38	22	PIT	Single interrupt vector for all channels
0x0000_009C	39	23	LTC(AESA)	

Table continues on the next page...

Table 4-6. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	Source module	Source description
0x0000_00A0	40	24	2.4G Radio INT0	Selectable interrupt request from BTLE or Generic FSK
0x0000_00A4	41	25	• — (for KW35) • FlexCAN0 (for KW36)	FlexCAN0 ORed Message Buffer (MB0~31) Interrupt (for KW36)
0x0000_00A8	42	26	2.4G Radio INT1	Selectable interrupt request from BTLE or Generic FSK
0x0000_00AC	43	27	MCG	
0x0000_00B0	44	28	LPTMR0	
0x0000_00B4	45	29	SPI1	Single interrupt vector for all sources
0x0000_00B8	46	30	Port Control Module	Pin detect (Port A)
0x0000_00BC	47	31	Port Control Module	Pin detect (Single interrupt vector for Port B and Port C)

1. Indicates the NVIC's interrupt source number.

4.3.2.3.1 Determining the bitfield and register location for configuring a particular interrupt

Suppose you need to configure the SPI0 interrupt. The following table is an excerpt of the SPI0 row from Interrupt channel assignments.

Table 4-7. Interrupt vector assignments

Address	Vector	IRQ[1]	NVIC IPR register number[2]	Source module	Source description
0x0000_0068	26	10	2	SPI0	Single interrupt vector for all sources

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.

- The NVIC registers you would use to configure the interrupt are:
  - NVICIPR2
- To determine the particular IRQ's bitfield location within these particular registers:
  - NVICIPR2 bitfield starting location = 8 * (IRQ mod 4) + 6 = 22

  Since the NVICIPR bitfields are 2-bit wide (4 priority levels), the NVICIPR2 bitfield range is 22-23

Therefore, the following bitfield locations are used to configure the SPI0 interrupts:

- NVICIPR2[23:22]

4.3.2.4 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.

4.3.3 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**.

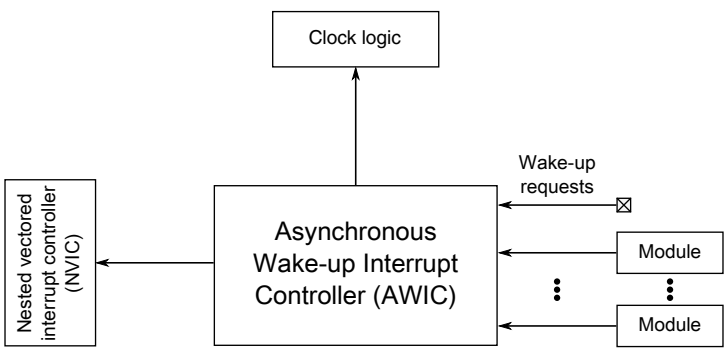


Figure 4-3. Asynchronous wake-up interrupt controller configuration

Table 4-8. 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

4.3.3.1 Wake-up sources

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

Table 4-9. 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	BLE or G-FSK interrupts
I^2Cx	Address match wakeup
LPUARTx	Any interrupt provided clock remains enabled
RTC	Alarm or seconds interrupt
NMI	NMI pin
TPMx	Any interrupt provided clock remains enabled
LPTMR	Any interrupt provided clock remains enabled
FlexCAN	FlexCAN Wake-up (KW36 only)

4.4 System modules

4.4.1 SIM configuration

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

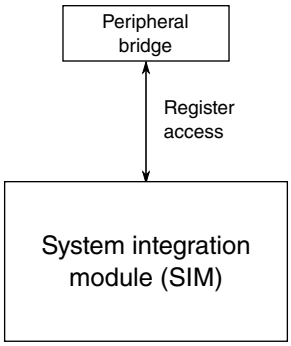


Figure 4-4. SIM configuration

Table 4-10. 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

4.4.2 System mode controller (SMC) configuration

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

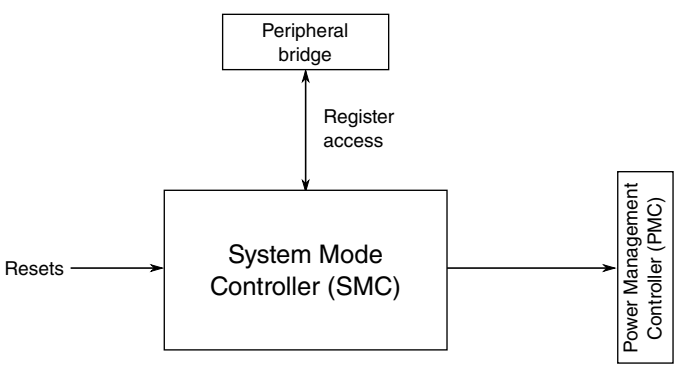


Figure 4-5. System mode controller configuration

Table 4-11. 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

Table continues on the next page...

Table 4-11. Reference links to related information (continued)

Topic	Related module	Reference
—	Power management controller (PMC)	PMC
	Low-leakage wakeup unit (LLWU)	LLWU
—	Reset control module (RCM)	Reset

4.4.3 PMC configuration

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

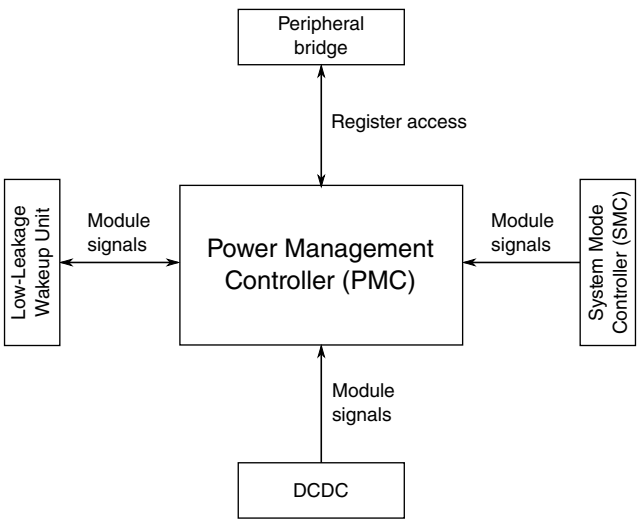


Figure 4-6. PMC configuration

Table 4-12. 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

4.4.4 DCDC configuration

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

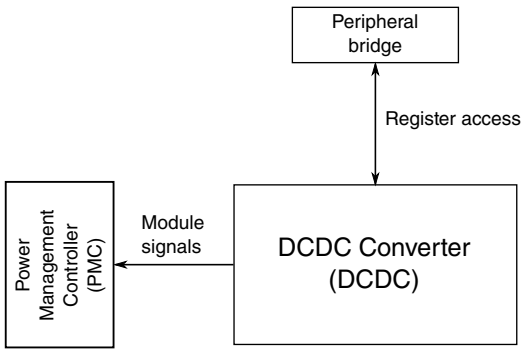


Figure 4-7. DCDC configuration

Table 4-13. Reference links to related information

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

The power architecture for this device is based on the PMC and a DCDC converter which can operate in Buck or Bypass mode.

4.4.4.1 Buck Mode

Buck mode supports operation over a wide range of battery voltages. See the VDD_{DCDC_IN} specification in the data sheet. After DCDC startup, the DCDC can continue to operate if the battery voltage drops as low as 1.8V

The figure below illustrates the power tree when the DCDC is configured in Buck mode.The power pins shown in the figure are used as follows:

- The DCDC_CFG pin is connected to V_{DCDC_IN} to select Buck configuration.
- The PSWITCH pin is used to power-on the DCDC. A push-button switch can be used with PSWITCH to momentarily ($T_{DCDC\_ON\_BUCK}$) connect PSWITCH to V_{DCDC_IN} to power-on the device, though in some use cases PSWITCH could be shorted to V_{DCDC_IN} or used with a different type of switch. If PSWITCH is used with a switch as shown in the figure below, an external pull-down resistor is needed on PSWITCH to ensure that it is 0 when the switch is off.

- V_{DCDC_IN} powers the DCDC regulator. It is expected that a capacitor is connected to V_{DCDC_IN} to help support large transient or short duration loads, including Rx and Tx bursts.
- VDD_1P5 is the 1.5V output of the DCDC. It needs to be filtered off-chip.
- VDD_PMC is a pad which is connected to VDD_1P5 in the package substrate. This pad supplies power to the PMC's LDO regulator which generates 1.2V to the internal logic.
- VDD_RF1/VDD_RF2/VDD_RF3 supplies power to the RF-analog domain regulators. It is powered from a filtered version of the 1.5V DCDC output
- VDD_1P8 is the second output of the DCDC. At DCDC power-up, this defaults to 1.8V. However, it can be programmed after power-up to provide a higher voltage, up to V_{DCDC_IN}. VDD_1P8 is expected to be filtered off-chip with a capacitor.
- VDD (I/O and DGO) supplies power to the digital I/O pins and the DGO logic that is always powered whenever the DCDC is enabled.
- VDDA powers the SAR ADC. This can be connected to VDD_1P8.
- VREFH is a reference option for the SAR ADC. This can be connected to VDD_1P8 or an external reference can be used.

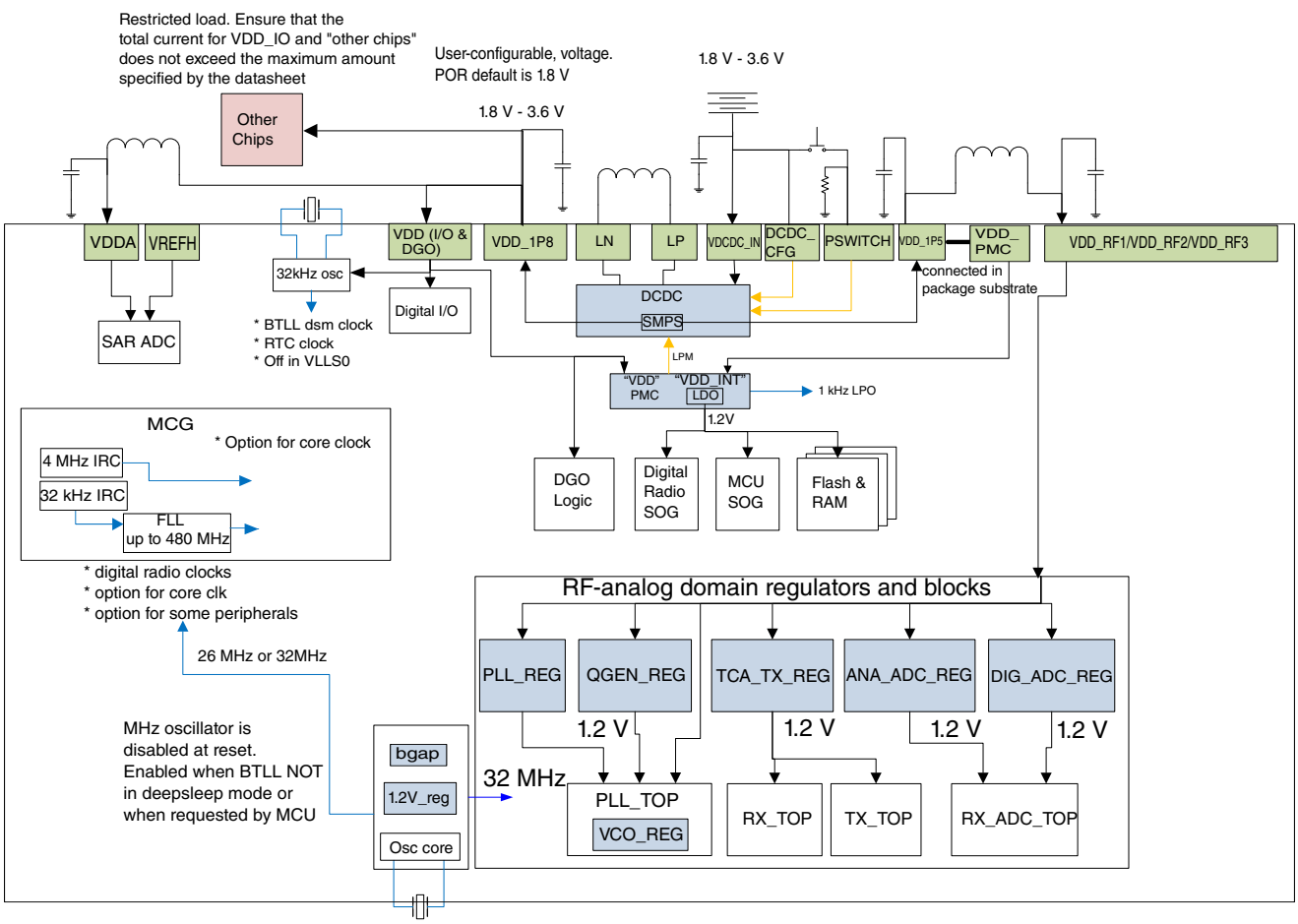


Figure 4-8. Buck Power Tree

The figure illustrates the use case where a push-button switch is used with PSWITCH pin. When the DCDC is off, depressing the button would cause the PSWITCH to be asserted momentarily to V_{DCDC_IN}, which would turn on the DCDC. After power-on, software can also check the state of the PSWITCH input; this could be used for example to signal that the device should be powered-off under software control through appropriate DCDC registers. The device does not require that a push-button be used; PSWITCH could be shorted to V_{DCDC_IN} (in which case the DCDC would always remain on while the battery is present) or used with a different type of switch.

NOTE

When using a rechargeable battery the PSWITCH pin should not be shorted to V_{DCDC_IN}, as leaving the DCDC always enabled may damage rechargeable batteries due to over-discharge.

It is expected that software will monitor the V_{DCDC_IN} periodically (using the SAR ADC), adjust the DCDC settings as needed to optimize performance, and shut off the DCDC if the battery voltage becomes too low.

Depending on the use case and low power mode, it may be more efficient to use the pulsed mode of the DCDC instead of continuous mode. The mapping of the device low power modes to the DCDC mode is shown in the table below.

Table 4-14. Low Power Configuration for Buck Mode

Device Low Power Mode	DCDC mode
RUN, WAIT and STOP modes	continuous mode
VLPR and VLPW modes	user configurable: continuous, or pulsed mode
VLPS mode	user configurable: continuous, or pulsed mode
LLSx and VLLSx modes	pulsed mode

On exit from pulsed mode to continuous mode, there is an expected delay ($T_{DCDC\_SETTLE\_BUCK}$) before the DCDC is fully capable of supporting the maximum loads. This is not expected to impact VDD_1P5 with an appropriately sized external capacitor, but the user may need to delay enabling large loads on VDD_1P8 (digital I/O, or external circuits). This delay should be completely hidden for exit from VLLSx modes, but not for LLSx or VLPx modes, so the users will need to be aware of this restriction.

4.4.4.2 Bypass Mode

The figure below illustrates the power tree when the DCDC is configured in Bypass mode.The power pins shown in the figure are used as follows:

- The DCDC_CFG pin is tied to HIGH and the PSWITCH pin is tied to ground (GND) to select bypass mode operation.
- The VDCDC_IN needs to be powered to allow the DCDC to recognize the DCDC_CFG and PSWITCH signals.
- LN, LP, and VDD_1P8 pins are not used.
- VDD_PMC is a 1.5-3.6 V input which supplies power to the PMC's LDO regulator which generates 1.2 V to the internal logic.
- VDD_RF1/VDD_RF2/VDD_RF3 is a 1.5-3.6 V input to the the RF-analog domain regulators.
- VDD is a 1.71-3.6V input which supplies power to the digital I/O pins and the DGO logic that is always powered whenever the DCDC is enabled.
- VDDA powers the SAR ADC. Vdda needs to be within +/- 100mV of VDD
- VREFH is a reference option for the SAR ADC.

Bypass mode

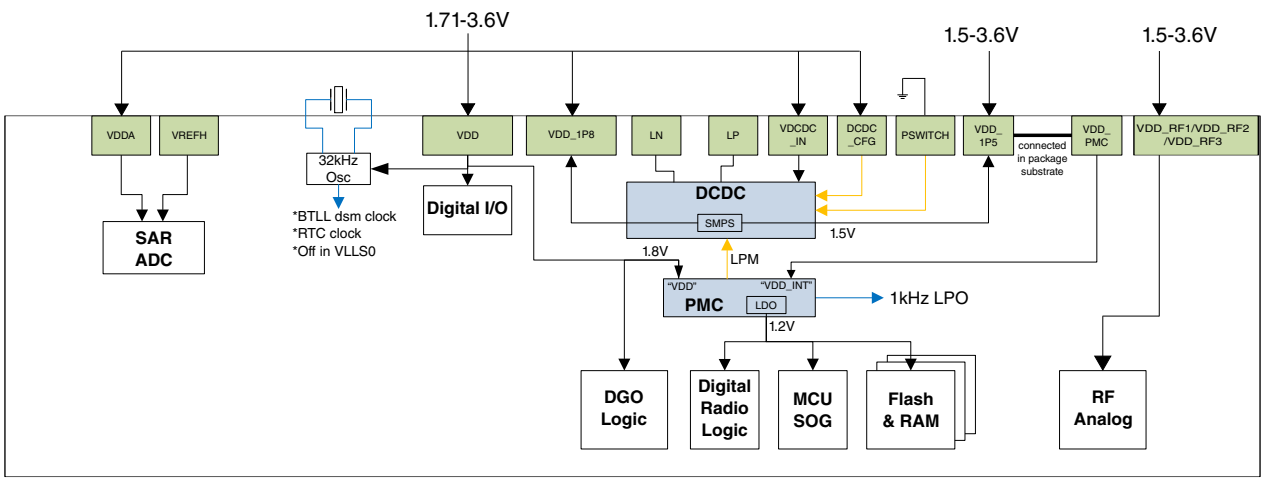


Figure 4-9. Bypass Power Tree

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

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

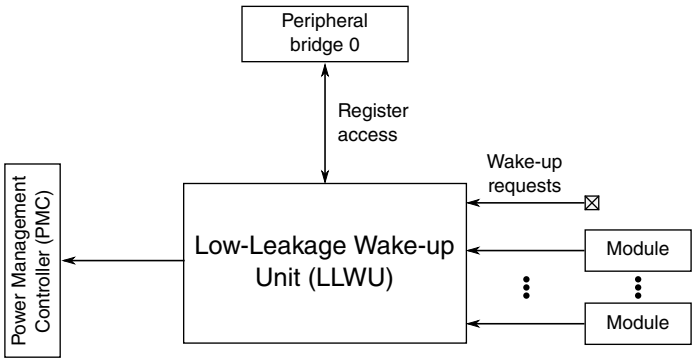


Figure 4-10. Low-Leakage Wake-up Unit configuration

Table 4-15. 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
	Power Management Controller (PMC)	Power Management Controller (PMC)
	System Mode Controller (SMC)	System Mode Controller
Wake-up requests		LLWU wake-up source

4.4.5.1 LLWU interrupt

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.

4.4.5.2 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 4-16. 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	Reserved
LLWU_P10	PTC2[1]
LLWU_P11	PTC3[1]
LLWU_P12	PTC4[1]
LLWU_P13	PTC5[1]
LLWU_P14	PTC6[1]
LLWU_P15	PTC7[1]
LLWU_M0IF	LPTMR0
LLWU_M1IF	CMP0
LLWU_M2IF	Generic Radio Wake up Request[2]
LLWU_M3IF	DCDC[3]
LLWU_M4IF	Reserved
LLWU_M5IF	RTC Alarm
LLWU_M6IF	Reserved
LLWU_M7IF	RTC Seconds

1. When the DCDC is operating in buck 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. BLE link layer controller and Generic FSK link layer controller can both generate the wake up signal.
3. This is for the PSWITCH interrupt

4.4.6 MCM configuration

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

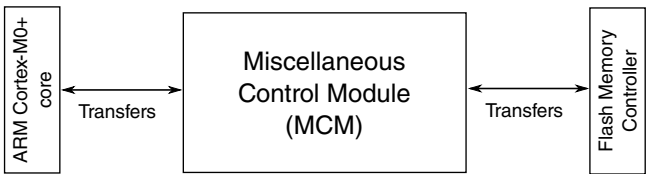


Figure 4-11. MCM configuration

Table 4-17. 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

4.4.7 Crossbar-light switch configuration

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

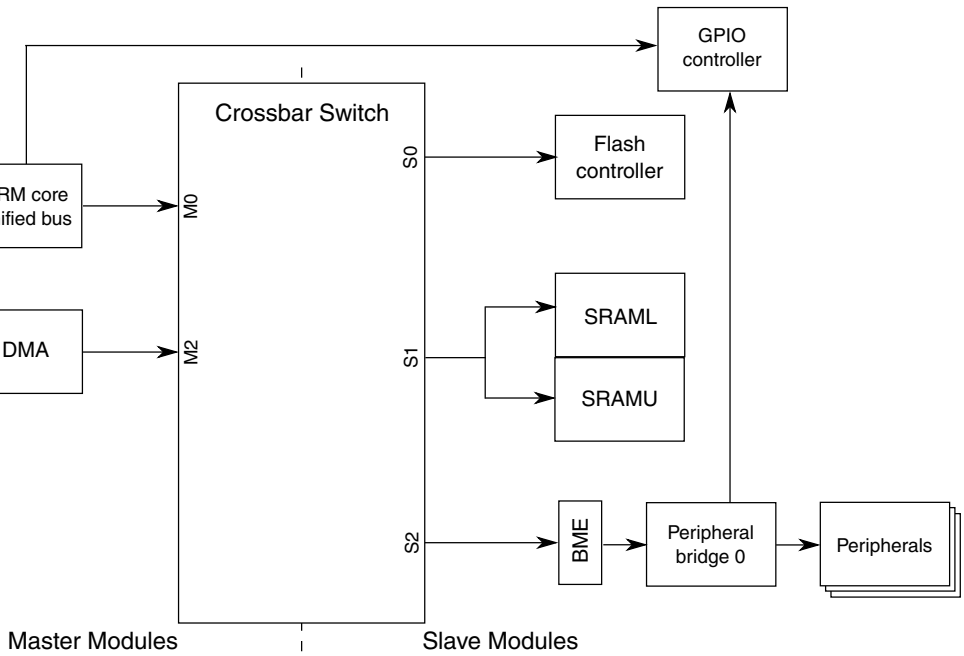


Figure 4-12. Crossbar-light switch integration

Table 4-18. 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

In this device, Crossbar-light (AXBS) supports both fixed-priority and round-robin slave port arbitration. For control bits, see MCM chapter.

4.4.7.1 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

4.4.7.2 Crossbar Switch Slave Assignments

This device contains 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

4.4.8 Peripheral bridge configuration

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

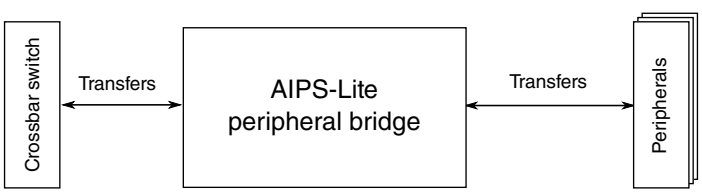


Figure 4-13. Peripheral bridge configuration

Table 4-19. 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

4.4.8.1 Number of peripheral bridges

This device contains one peripheral bridge.

4.4.8.2 Memory maps

The peripheral bridges are used to access the registers of most of the modules on this device. See Peripheral Bridge (AIPS-Lite) Memory Map for the memory slot assignment for each module.

4.4.9 DMA request multiplexer configuration

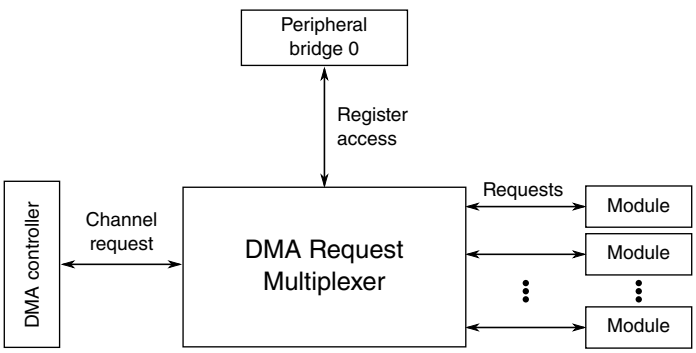


Figure 4-14. DMA request multiplexer configuration

Table 4-20. 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

4.4.9.1 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 4-21. DMA request sources - MUX 0

Source number	Source module	Source description	Async DMA capable
0	—	Channel disabled[1]	
1	Reserved	Not used	
2	LPUART0	Receive	Yes
3	LPUART0	Transmit	Yes
4	LPUART1	Receive	Yes
5	LPUART1	Transmit	Yes
6	Reserved	—	
7	Reserved	—	
8	Reserved (for KW35) FlexCAN0 (for KW36)	— (for KW35) FlexCAN0 (for KW36)	
9	Reserved	—	
10	Reserved	—	
11	2.4G Radio	GFSK DMA or XCVR DMA	
12	Reserved	—	
13	Reserved	—	
14	Reserved	—	
15	Reserved	—	
16	SPI0	Receive	
17	SPI0	Transmit	
18	SPI1	Receive	
19	SPI1	Transmit	
20	LTC (AESA)	Input FIFO	
21	LTC (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	—	

Table continues on the next page...

Table 4-21. DMA request sources - MUX 0 (continued)

Source number	Source module	Source description	Async DMA capable
38	Reserved	—	
39	Reserved	—	
40	ADC0	—	Yes
41	Reserved	—	
42	CMP0	—	Yes
43	Reserved	—	
44	Reserved	—	
45	Reserved	—	
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	Reserved	—	
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.

4.4.9.2 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 .

4.4.10 DMA Controller Configuration

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

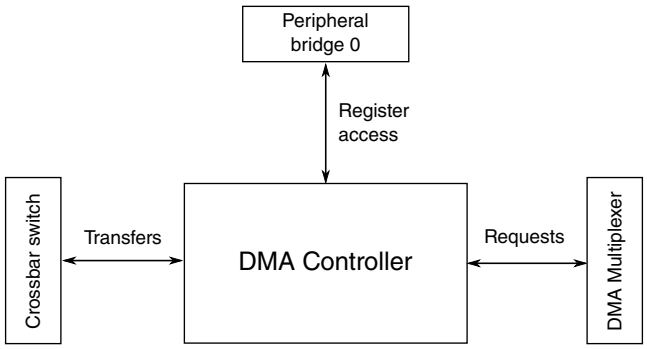


Figure 4-15. DMA Controller configuration

Table 4-22. 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

4.4.11 Computer operating properly (COP) watchdog configuration

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

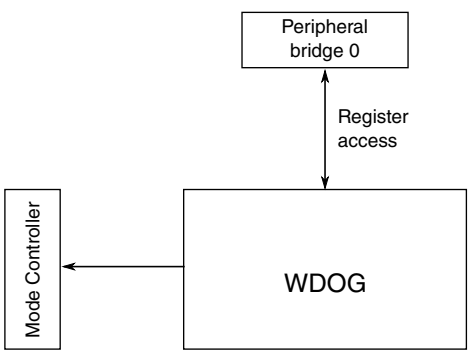


Figure 4-16. COP watchdog configuration

Table 4-23. Reference links to related information

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

Table continues on the next page...

Table 4-23. Reference links to related information (continued)

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

4.4.11.1 COP clocks

The multiple clock inputs for the COP are:

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

4.4.11.2 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_COPC[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_COPC[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 4-24. 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 COPSTPEN 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.

4.4.11.3 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.

4.5 Clock modules

4.5.1 MCG configuration

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

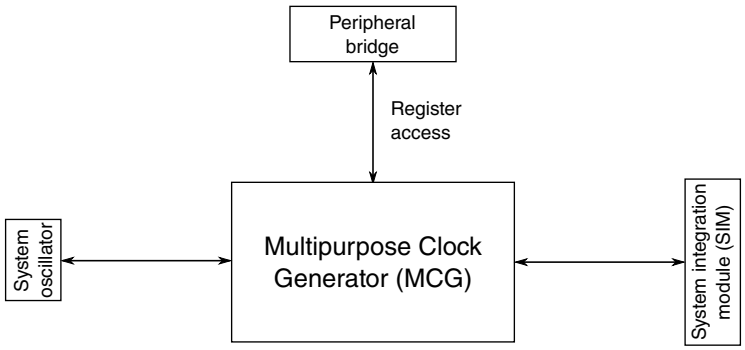


Figure 4-17. MCG configuration

Table 4-25. 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

4.5.1.1 MCG Instantiation Information

This device supports one MCG module.

- OSC0 refers to the reference (26 MHz or 32 MHz) oscillator
- MCG_C2[RANGE] should be set to 1x for clock monitor function to work correctly with the RF oscillator
- MCG_C2[HGO] is not used
- MCG_C2[EREFS] is not used
- MCG_S[OSCINIT0] is tied to 0 in KW36/35. 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

4.5.1.2 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.

4.5.2 32kHz OSC Configuration

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

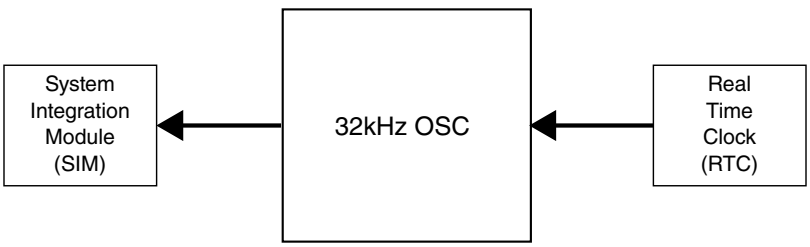


Figure 4-18. 32kHz OSC configuration

Table 4-26. 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

4.5.2.1 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

4.5.3 Reference Oscillator Configuration

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

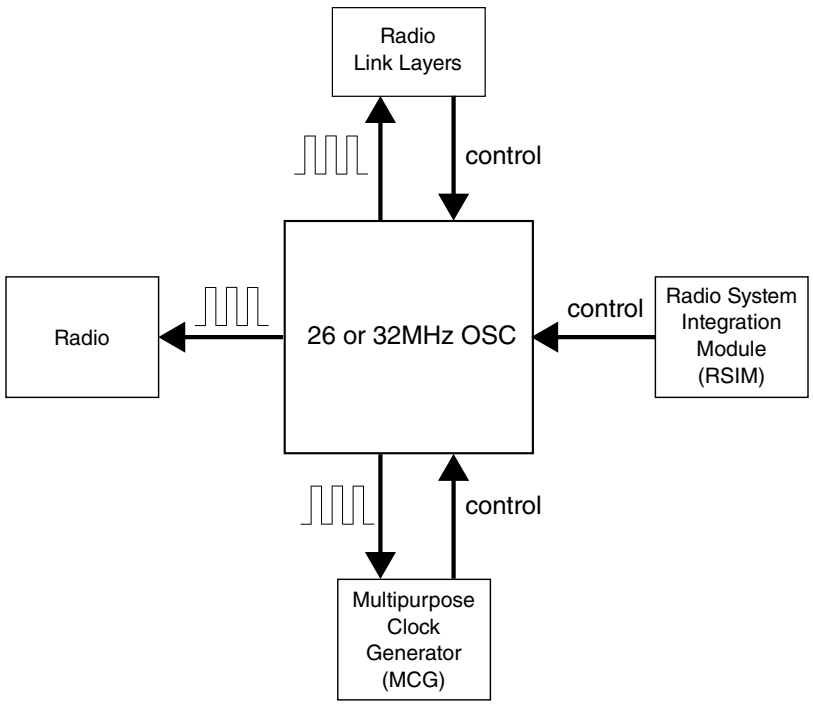


Figure 4-19. Reference OSC configuration

Table 4-27. Reference links to related information

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

4.5.3.1 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 Radio Link Layer, Radio System Integration Module (RSIM) and Multipurpose Clock Generator (MCG).

The radio link layers enable the reference oscillator when they are not in deepsleep mode (DSM).

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.

4.6 Memories and memory interfaces

4.6.1 Flash Memory Configuration

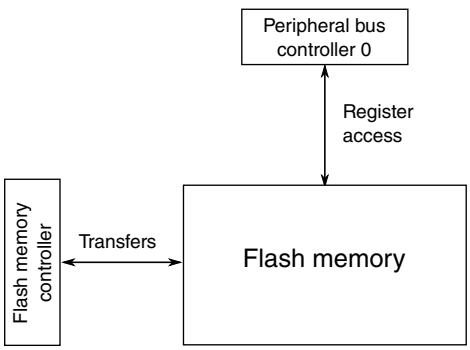


Figure 4-20. Flash memory configuration

Table 4-28. 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

4.6.1.1 Flash Memory Sizes

KW35A/Z contains two sub family products: KW35 and KW36. The flash memory details of KW35 is as follows:

- KW35 contains 512 KB flash, which is composed by two 256 KB Program flash (P-flash) array with 2 KB sectors.
- Flash address ranges from 0x0000 0000 to 0x0007 FFFF.

- All 2 x 256 KB P-flash can be protected by the Flash Access Control (FAC) feature.
- KW35 contains 8 KB Program acceleration RAM. The Program acceleration RAM address ranges from 0x1400 0000 to 0x1400 1FFF. When using Program Section command to program a whole flash section, the lower quarter of the Program acceleration RAM is supposed to buffer the data to be programmed to the target sector. When flash section programming is not needed, the whole RAM can serve as general purpose RAM.

The following figure shows the KW35 flash memory map.

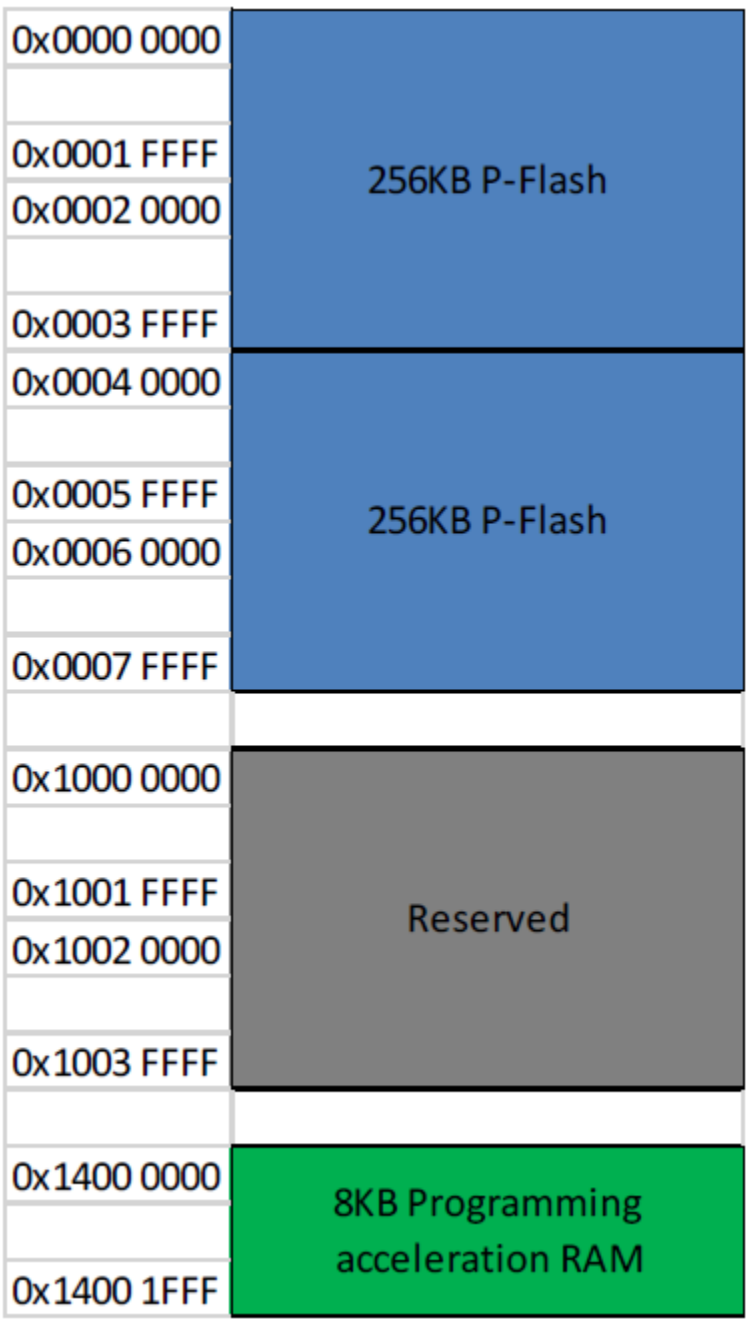


Figure 4-21. KW35 Flash Memory Map

The flash memory details of KW36 is as follows:

- KW36 contains one 256 KB program flash (P-flash) array with 2 KB sectors.
- Flash address ranges from 0x0000 0000 to 0x0003 FFFF. This 1 x 256 KB P-flash can be protected by the Flash Access Control (FAC) feature.
- KW36 contains one 256 KB FlexNVM array with address range from 0x1000 0000 to 0x1003 FFFF, and one 8 KB FlexRAM with address range from 0x1400 0000 to 0x1400 1FFF. With FlexRAM, user can configure FlexNVM as either
  - basic data flash
  - EEPROM flash records to support the built-in EEPROM feature, or
  - a combination of both
- If no EEPROM feature is required, the whole 256 KB FlexNVM can serve as basic data flash (D-flash).
- Besides address range starting from 0x1000 0000 to 0x1003 FFFF, D-flash also has an alias starting from address 0x0004 0000. D-flash cannot be protected by the Flash Access Control (FAC) feature.
- The 8 KB FlexRAM serve as 8 KB Program acceleration RAM. When flash section programming is not required, the whole RAM can serve as a general purpose RAM.

The following figure shows the KW36 flash memory map when 256 KB FlexNVM is partitioned as 256 KB D-flash with no EEPROM backup.

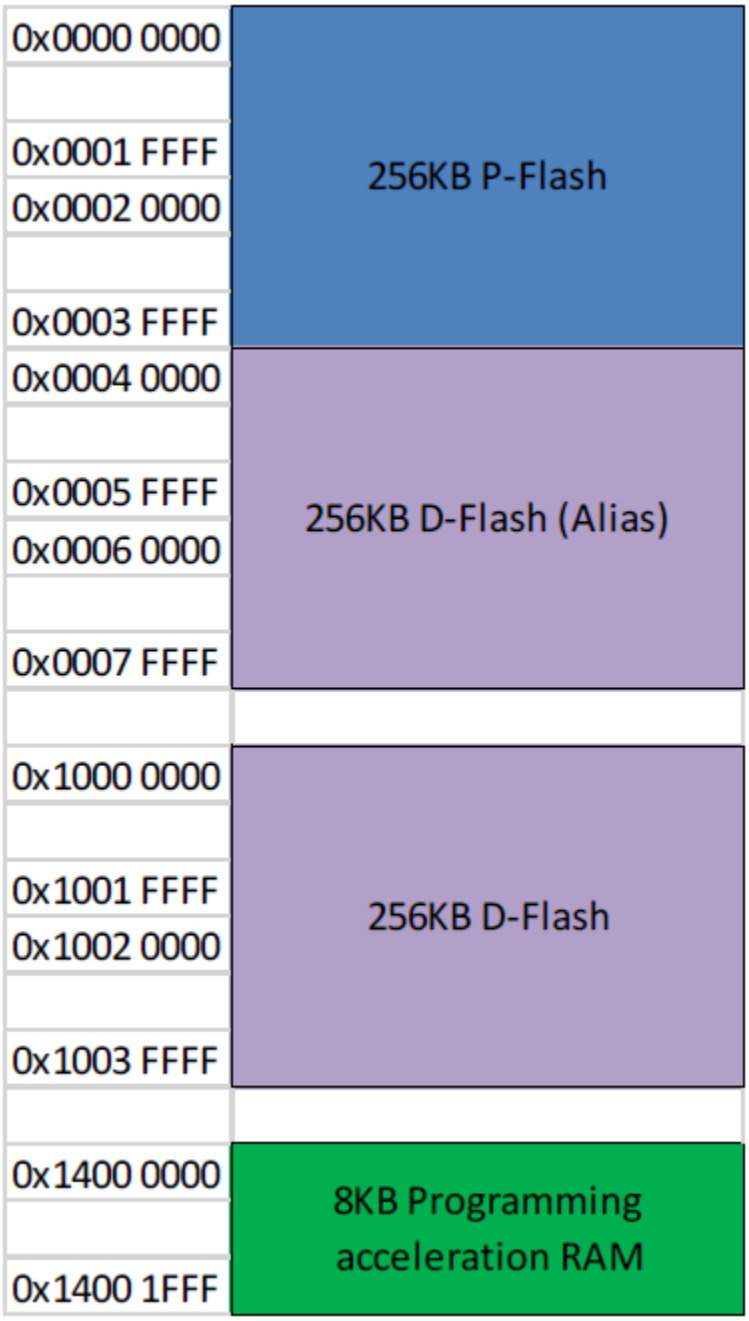


Figure 4-22. KW36 Flash Memory Map Example, 256 KB D-flash Partition

When FlexNVM is partitioned as D-flash - when programming D-flash by FTFE commands (Program Phrase command or Program Section command), user should set the 24-bit flash address MSB to '1' to distinguish D-flash from P-flash. For example, to program D-flash at address 0x1000 1000 (this address also has an alias at 0x0004 1000), user should use address 0x80 1000 as the Program Phrase command address field. If EEPROM feature is required, the 256 KB FlexNVM needs to be partitioned as basic D-flash and EEPROM backup. If 256 KB is not assigned as EEPROM backup, the remaining FlexNVM can still serve as D-flash with address range starting from 0x1000

0000. The end address depends on the D-flash size partitioned. This D-flash also has an alias starting from address 0x0004 0000. D-flash cannot be protected by the Flash Access Control (FAC) feature. The FlexRAM serves as EEPROM access interface with address range starting from 0x1400 0000. The end address depends on the EEPROM size configured. The following figure shows the KW36 flash memory map when 256 KB FlexNVM is partitioned as 128 KB D-flash, 128 KB EEPROM backup, and 8 KB EEPROM.

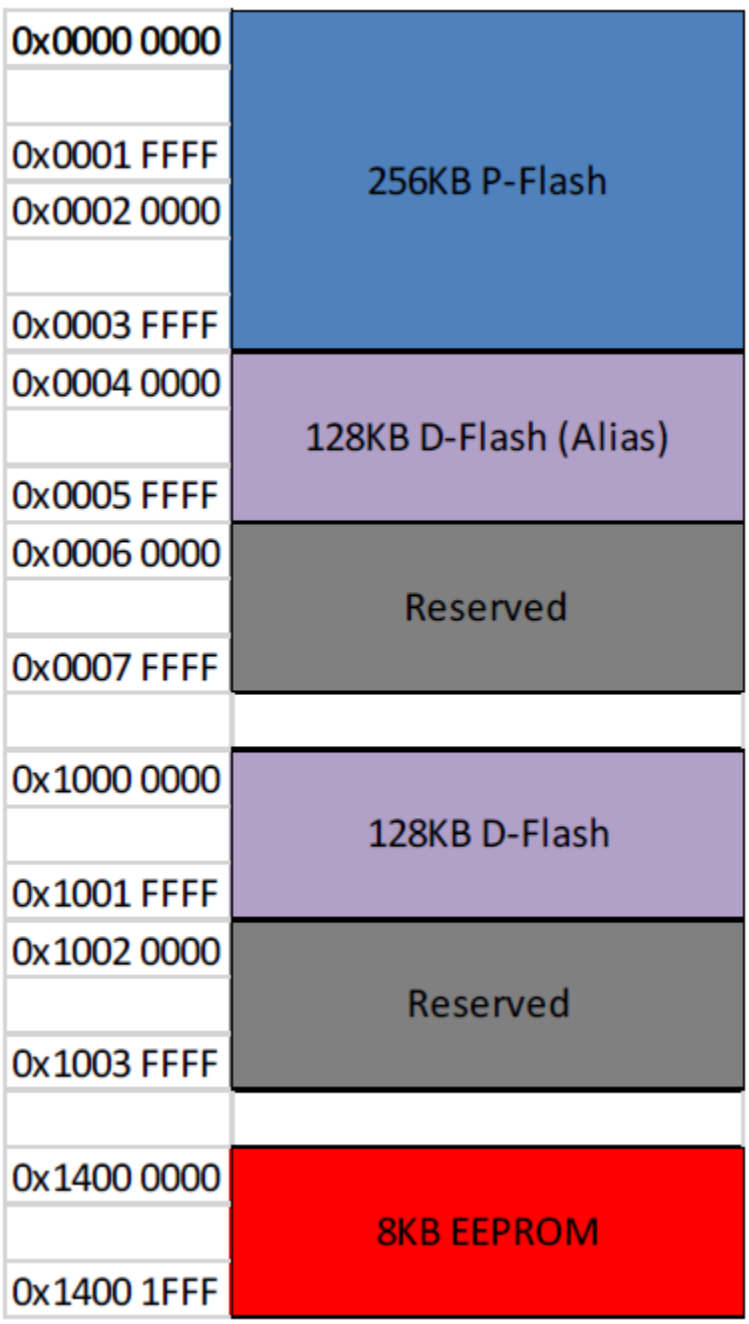


Figure 4-23. KW36 Flash Memory Map, 128 KB D-flash and 128 KB EEPROM Backup Partition

The following figure shows the KW36 flash memory map when 256 KB FlexNVM is partitioned as no D-flash, but as 256 KB EEPROM backup and 8KB EEPROM.

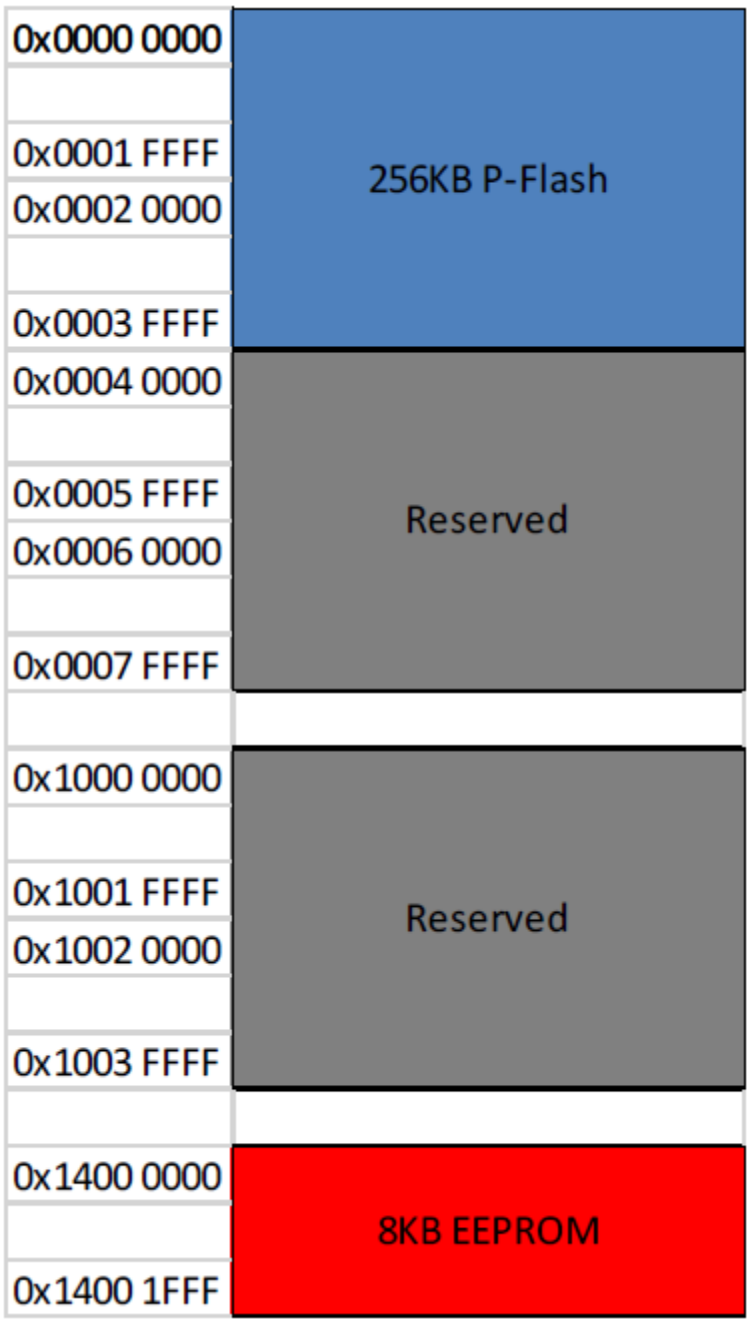


Figure 4-24. KW36 Flash Memory Map, 256 KB EEPROM Backup Partition

4.6.1.2 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.

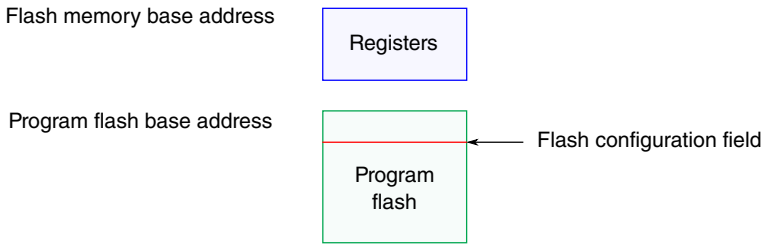


Figure 4-25. Flash memory map

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.

4.6.1.3 Flash Security

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

4.6.1.4 Flash Modes

The FTFE 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.

4.6.1.5 Erase All Flash Contents

An Erase All Flash Blocks operation can be launched by software through a series of peripheral bus writes to flash registers. 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.

4.6.1.6 FTFE_FOPT Register

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

4.6.2 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.

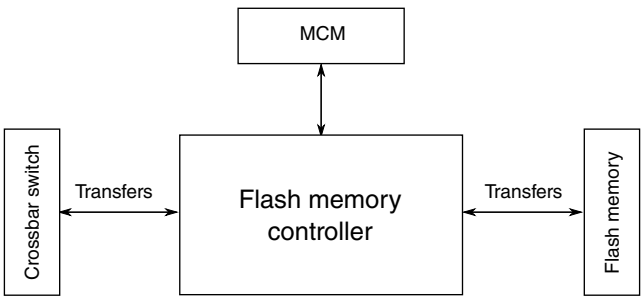


Figure 4-26. Flash memory controller configuration

Table 4-29. 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

4.6.3 SRAM Configuration

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

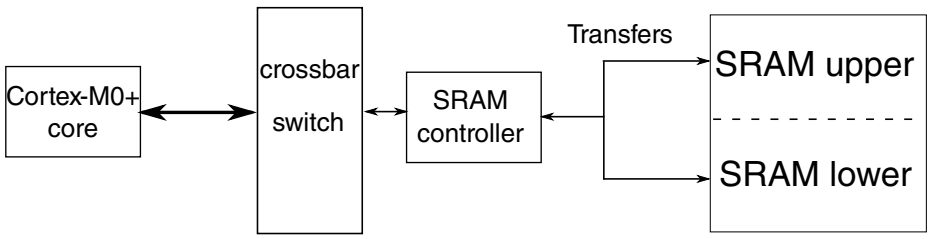


Figure 4-27. SRAM configuration

Table 4-30. 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

4.6.3.1 SRAM Sizes

The device contains 64 KB of SRAM which can be accessed by bus masters through the cross-bar switch.

4.6.3.2 SRAM Ranges

The device contains 64 KB of SRAM, split into two ranges, 1/4 (16 Kbytes) is allocated to SRAM_L and 3/4 (48 Kbytes) is allocated to SRAM_U. The first 16 Kbytes of SRAM_U remains powered in LLS2 and VLLS2 modes. The ranges are as follows:

- SRAM_L: 0x1FFF_C000-0x1FFF_FFFF (16 Kbytes). Powered off in LLS2 and VLLS2
- SRAM_U: 0x2000_0000-0x2000_BFFF (48 Kbytes)
  - 0x2000_0000 to 0x2000_3FFF (16 Kbytes). Remains powered in LLS2 and VLLS2
  - 0x2000_4000 to 0x2000_7FFF (16 Kbytes). Powered off in LLS2 and VLLS2. This can be powered on by setting SMC_STOPCTRL[RAM2PO]
  - 0x2000_8000~0x2000_BFFF (16 Kbytes): Powered off in LLS2 and VLLS2

This is illustrated in the following figure.

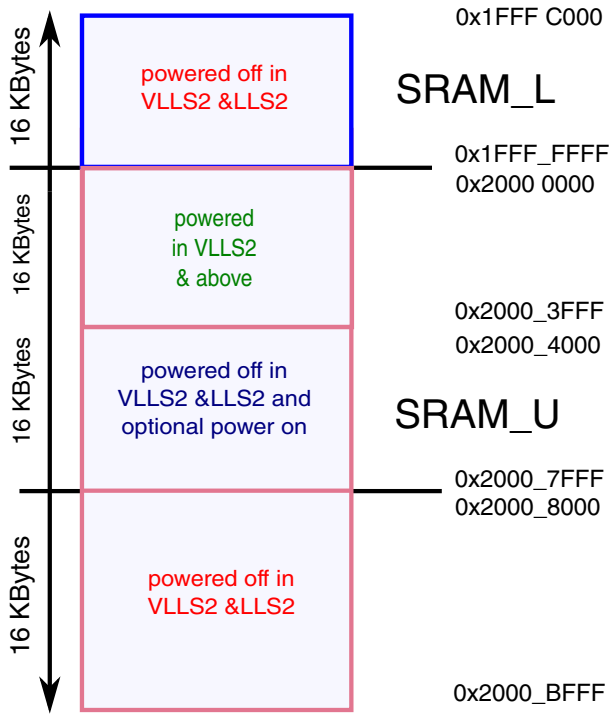


Figure 4-28. SRAM blocks memory map

4.6.3.3 SRAM retention in low power modes

In VLLS1 and VLLS0, no SRAM is retained. In LLS2 and VLLS2 modes, the 16 KB region of SRAM_U range from 0x2000_0000 to 0x2000_3FFF is powered. Optionally another 16 KB region of SRAM_U range from 0x2000_4000 to 0x2000_7FFF can be powered on, enabled by setting SMC_STOPCTRL[RAM2PO]. In other low power modes the contents of the SRAM are retained.

4.6.4 System Register File Configuration

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

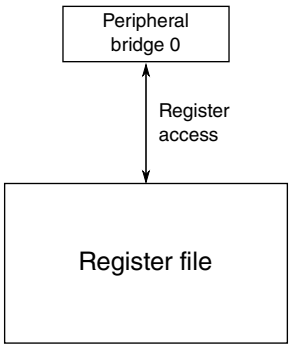


Figure 4-29. System Register file configuration

Table 4-31. 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

4.6.4.1 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.

4.7 Security

4.7.1 TRNG configuration

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

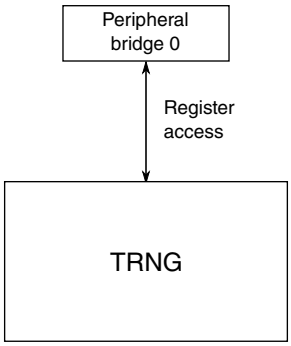


Figure 4-30. MCG configuration

Table 4-32. Reference links to related information

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

4.7.2 LTC configuration

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

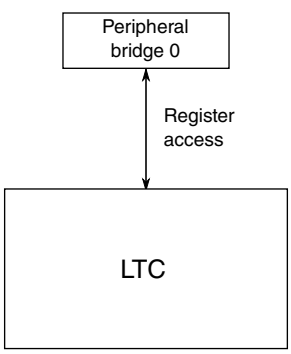


Figure 4-31. LTC configuration

Table 4-33. Reference links to related information

Topic	Related module	Reference
Full description	LTC	LTC
System memory map	—	System memory map

Table continues on the next page...

Table 4-33. Reference links to related information (continued)

Topic	Related module	Reference
Clocking	—	Clock distribution
Power management	—	Power management
Signal multiplexing	Port control	Signal Multiplexing

4.8 Analog

4.8.1 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 48pin package VREFH and VDDA are separate package pins, while VREFL is connected to the VSSA package pin. VREF module output VREF_OUT share the same pin with VREFH. When VREF is enabled, the VREF output serve as the reference voltage. When VREF is disabled, external voltage reference is used and input via VREFH pin.

Table 4-34. 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
CMP with 6-bit DAC	0 (Vin1) - VREFH 1 (Vin2) - VDD	Selected by CMPx_DACCR[VRSEL] bit

4.8.2 16-bit SAR ADC configuration

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

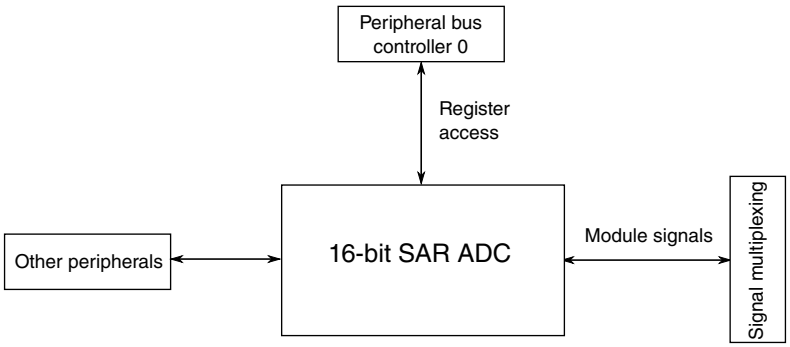


Figure 4-32. 16-bit SAR ADC configuration

Table 4-35. 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

4.8.2.1 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 two SC1n (Status and Control) registers, ADC0_SC1A and ADC0_SC1B, and the corresponding result registers.

The asynchronous hardware conversion trigger of ADC, ADHWT, can be initiated by various peripherals like TPM, PIT, CMP, RTC, or Radio. See SIM_SOPT7[ADC0TRGSEL] for more details.

4.8.2.2 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.

4.8.2.3 ADC0 Connections/Channel Assignment

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

4.8.2.3.1 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
00101	AD5	Reserved	ADC0_SE5
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
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.

4.8.2.4 Alternate clock

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

4.8.3 CMP Configuration

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

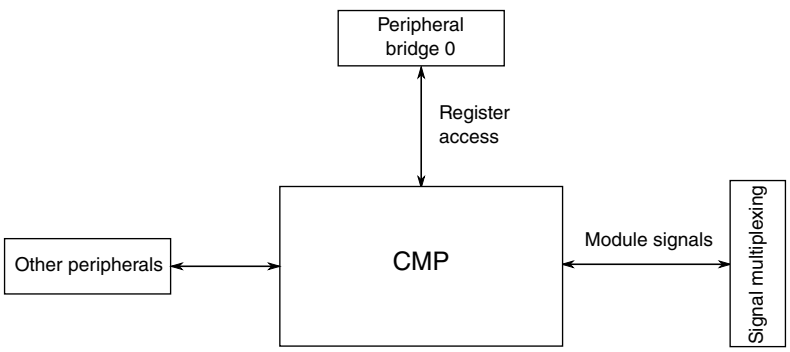


Figure 4-33. CMP configuration

Table 4-36. 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

4.8.3.1 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 CMP input connections 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 LPUART IR interfaces. For complete details on the CMP module interconnects, refer to the Module to Module Interconnects 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.

4.8.3.2 CMP input connections

The following table shows the CMP input channel assignments.

Table 4-37. CMP input connections

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

4.8.3.3 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

4.8.3.4 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, the LPTMR provides control for this two-staged sequencing. 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.

4.8.4 Voltage Reference (VREF)

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

Table 4-38. Reference links to related information

Topic	Related module	Reference
Full description	Voltage Reference	Voltage Reference(VREF1)
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

4.8.4.1 VREF Overview

This device includes a voltage reference (VREF) to supply an accurate 1.2 V voltage output.

The voltage reference can provide a reference voltage to external peripherals or a reference to analog peripherals, such as the ADC, DAC, or CMP.

NOTE

PMC_REGSC[BGEN] bit must be set if the VREF regulator is required to remain operating in VLPx modes.

NOTE

For either an internal or external reference if the VREF_OUT functionality is being used, VREF_OUT signal must be connected to an output load capacitor. Refer the device data sheet for more details.

4.9 Radio

4.9.1 Radio module configuration

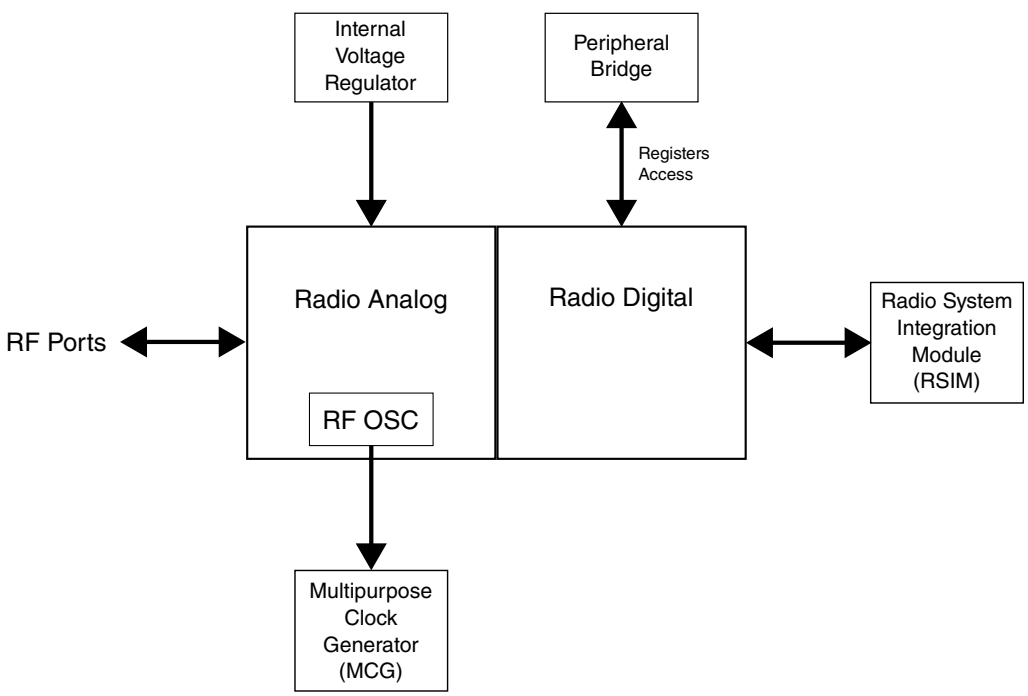


Figure 4-34. Radio configuration

Table 4-39. 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

Table continues on the next page...

Table 4-39. Reference links to related information (continued)

Topic	Related module	Reference
Power management	—	Power management
Signal multiplexing	Port control	Signal Multiplexing

4.9.1.1 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 KW36/35, 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.

4.10 Timers

4.10.1 Timer/TPM Configuration

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

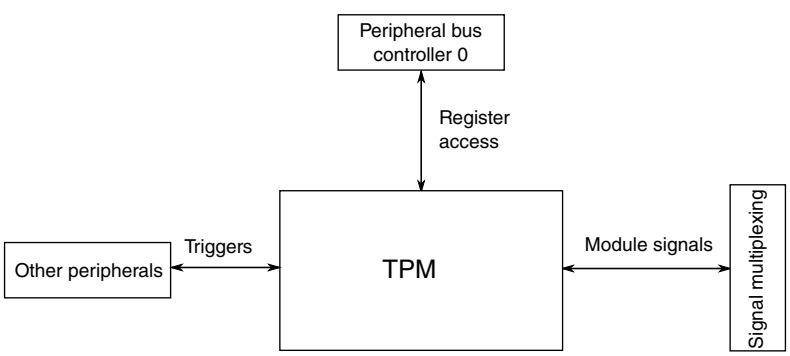


Figure 4-35. TPM configuration

Table 4-40. Reference links to related information

Topic	Related module	Reference
Full description	Timer/PWM Module	Timer/PWM Module
System memory map		System memory map

Table continues on the next page...

Table 4-40. Reference links to related information (continued)

Topic	Related module	Reference
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

4.10.1.1 TPM Instantiation Information

This device contains three Low Power TPM modules (TPM). TPM0 is configured as shown in the table below. All TPMs can be functional in Stop/VLPS mode; the clock source is either external or internal in Stop/VLPS mode.

Table 4-41. TPM configuration

Instance Name	Channels	Features
TPM0	4	• Basic TPM • Quadrature decoder and filtering are not supported. FILTER, COMBINE, and QDCTRL registers do not exist. • Functional in Stop/VLPS mode
TPM1	2	• Basic TPM • Quadrature Decoder and filtering to support the wireless mouse use case • Functional in Stop/VLPS • Does not support C2SC, C3SC, C2V, and C3V registers.
TPM2	2	• Basic TPM • Quadrature Decoder and filtering to support the wireless mouse use case • Functional in Stop/VLPS • Does not support C2SC, C3SC, C2V, and C3V registers.

4.10.1.2 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.

4.10.1.3 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 4-42. 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
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.

4.10.1.4 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.

4.10.1.5 Interrupts

The TPM has 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.

4.10.2 PIT Configuration

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

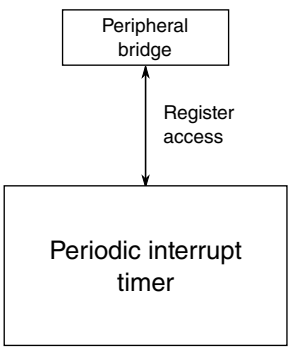


Figure 4-36. PIT configuration

Table 4-43. 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

4.10.2.1 PIT/DMA Periodic Trigger Assignments

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

Table 4-44. PIT channel assignments for periodic DMA triggering

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

4.10.2.2 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.

4.10.2.3 PIT/TPM Triggers

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

4.10.3 Low-power timer configuration

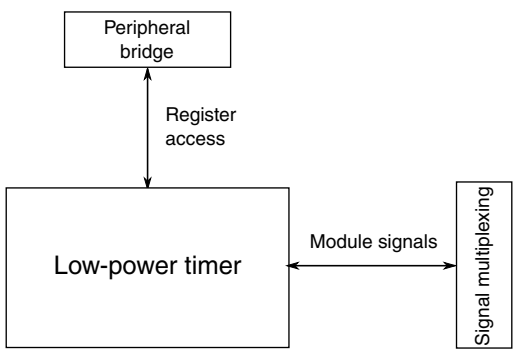


Figure 4-37. LPT configuration

Table 4-45. 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

4.10.3.1 LPTMR Instantiation Information

This device has one Low-power Timer (LPTMR) module. The 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.

4.10.3.2 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

4.10.3.3 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

4.10.4 RTC configuration

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

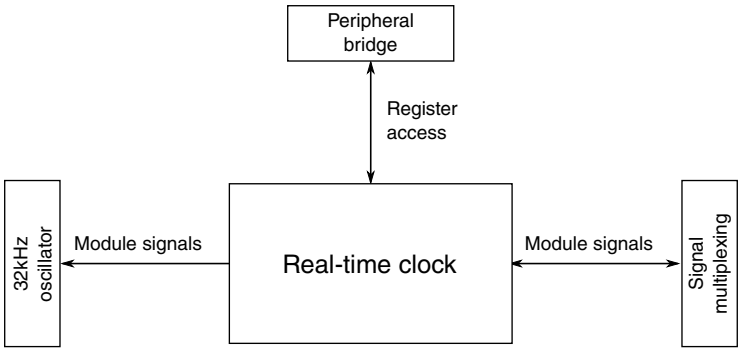


Figure 4-38. RTC configuration

Table 4-46. 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

4.10.4.1 RTC Instantiation Information

This device supports one RTC module. 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.

4.10.4.2 RTC_CLKOUT options

RTC_CLKOUT pin is driven with the RTC 1Hz output.

4.11 Communication interfaces

4.11.1 SPI configuration

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

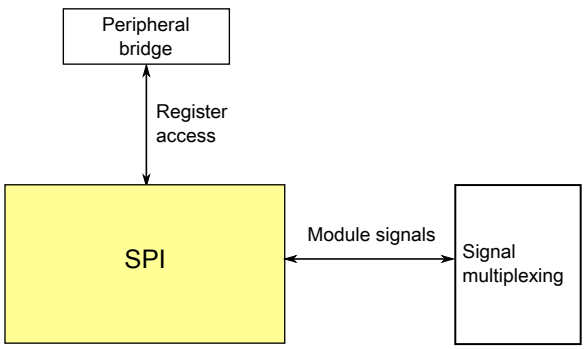


Figure 4-39. SPI configuration

Table 4-47. 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

4.11.1.1 SPI Instantiation Information

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

Both DSPI modules include a 4-deep FIFO and are clocked on the bus clock. SPI0 has three PCS signals while SPI1 has one.

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

Table 4-48. 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.

4.11.2 I2C Configuration

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

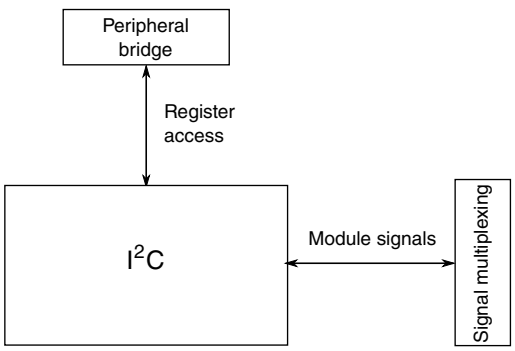


Figure 4-40. I2C configuration

Table 4-49. 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

4.11.2.1 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 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.

4.11.3 LPUART Configuration

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

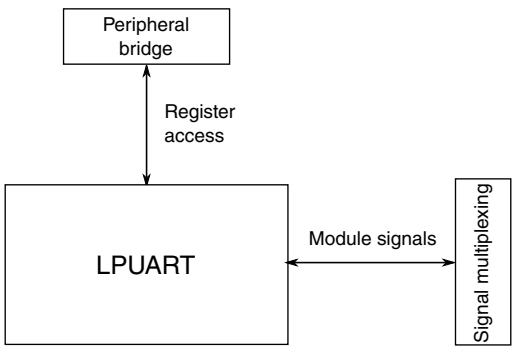


Figure 4-41. LPUART configuration

Table 4-50. Reference links to related information

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

4.11.3.1 LPUART instantiation information

KW35A/Z supports one LPUART module (LPUART0) and KW36A/Z supports two LPUART modules (LPUART0, LPUART1).

The LPUART0 and LPUART1 module supports basic UART with DMA interface function, x4 to x32 oversampling of baud-rate, and hardware flow control.

The LPUART0 and LPUART1 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.

4.11.4 FlexCAN configuration

4.11.4.1 FlexCAN instantiation information

The device includes a CAN (FlexCAN) module that supports the Flexible Data-rate (CAN FD) feature. This module is supported in KW36 only.

Table 4-51. FlexCAN Instantiation Information

Module	Number of Message Buffers	CAN FD feature
FlexCAN	32	Yes

FlexCAN clock can be the bus clock or the OSCERCLK.

The maximum bit rate can be configured as follows:

Table 4-52. Maximum Rate Configuration in FlexCAN

Mode	Protocol Engine (PE) source	Frequency	SYNC_SEG	PSEG1	PSEG2	Bit Rate	Register configure
Normal/FD address phase	OSC	16M	1	5	2	2 Mbps	CAN_CTRL1
FD data phase	OSC	16M	1	2	2	3.2 Mbps	CAN_FDCBT1

4.11.4.2 FlexCAN wake-up glitch filter configuration

FlexCAN can apply a low-pass glitch filter to protect the Rx CAN input from spurious wake-up. When CAN_MCR[WAKSRC] is set, FlexCAN uses the filtered Rx input to detect recessive to dominant edges on the CAN bus. The 4 MHz IRC serves as the glitch filter clock and an active IRCLK is required in STOP/VLPS mode by enabling MCG_C1[IRCLKEN] and MCG_C1[IREFSTEN] to be active on chip glitch filter.

4.11.4.3 FlexCAN free-running timer with external time tick

FlexCAN has a 16-bit free-running timer (CAN_TIMER) that can run with external time tick. When the TIMER_SRC field in CAN_CTRL2 is asserted, the timer is continuously incremented by an external time tick. This external time tick is connected with PIT channel 0 trigger output.

4.12 Human-machine interfaces (HMI)

4.12.1 GPIO Configuration

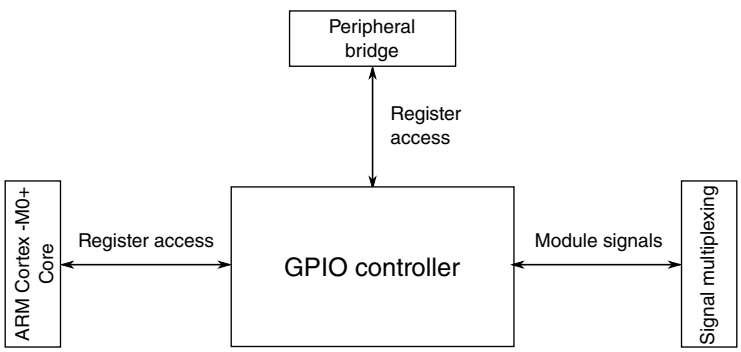


Figure 4-42. GPIO configuration

Table 4-53. 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

4.12.1.1 GPIO Instantiation Information

The device will include 9 pins with high current drive capability. These pins are PTC1, PTC2, PTC3, PTC6, PTC7,PTC17, PTC18, PTB0 and PTB1. High drive can be controlled for these pin using the PORTx_PCRn[DSE] field.

4.12.1.1.1 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.

4.12.1.2 Port control and interrupt summary

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

Table 4-54. 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=Enabled; Others=Disabled	PTB18 (NMI_b)=Enabled; Others=Disabled	Disabled
Slew rate enable control	Yes	Yes	Yes
Slew rate enable at reset	PTA0/PTA16/PTA17/PTA18/ PTA19=Disabled; Others=Enabled	PTB3/PTB0 = Disabled; Others=Enabled	PTC7/PTC16/PTC17/PTC18/ PTC19=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	PTC1/PTC2/PTC3/PTC6/ PTC7/PTC17/PTC18 only
Drive strength enable at reset	Disabled	Disabled	Disabled
Pin mux control	Yes	Yes	Yes
Pin mux at reset	PTA0/ PTA1=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. LPUART signals can be configured for open-drain using SIM_SOPT5 register. I2C signals are automatically enabled for open drain when selected.

4.12.1.3 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.

Chapter 5
Memory Map

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

5.1 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.

5.2 System memory map

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

Table 5-1. System memory map

System 32-bit Address Range	Destination Slave	Access
0x0000_0000–0x07FF_FFFF[1]	Program flash (KW35) FlexNVM Data flash alias (KW36)	All masters
0x0008_0000 – 0x0FFF_FFFF	Reserved	—
0x1000_0000 – 0x13FF_FFFF	Reserved (KW35) FlexNVM (KW36)	All masters
0x1400_0000 – 0x1FFF_BFFF	Programming Acceleration RAM (KW35) FlexRAM (KW36)	All masters
0x1FFF_C000 – 0x1FFF_FFFF	SRAM_L: Lower SRAM	All masters
0x2000_0000 – 0x2000_BFFF	SRAM_U: Upper SRAM[2]	All masters
0x2001_8000–0x3FFF_FFFF	Reserved	–
0x4000_0000–0x4007_FFFF	AIPS Peripherals	Cortex-M0+ core & DMA
0x4008_0000–0x400F_EFFF	Reserved	–

Table continues on the next page...

Table 5-1. System memory map (continued)

System 32-bit Address Range	Destination Slave	Access
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
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. See SRAM Ranges 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)

5.3 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)	MCG_C4[SCFTRIM]
0x0000_03FE (bit 4:1)	MCG_C4[FCTRIM]
0x0000_03FE (bit 6)	MCG_C2[FCFTRIM]
0x0000_03FF	MCG_C3[SCTRIM]

5.4 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.

5.5 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 (BME) for a detailed description of BME functionality.

5.6 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.

5.6.1 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.

5.6.2 Peripheral Bridge (AIPS-Lite) Memory Map

Table 5-2. 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	—

Table continues on the next page...

Table 5-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
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	FlexCAN (KW36 only)
0x4002_5000	37	—
0x4002_6000	38	—
0x4002_7000	39	—
0x4002_8000	40	—
0x4002_9000	41	TRNG
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	—

Table continues on the next page...

Table 5-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4003_3000	51	—
0x4003_4000	52	—
0x4003_5000	53	—
0x4003_6000	54	—
0x4003_7000	55	PIT
0x4003_8000	56	TPM0 (Timer/PWM)
0x4003_9000	57	TPM1
0x4003_A000	58	TPM2
0x4003_B000	59	ADC0
0x4003_C000	60	—
0x4003_D000	61	RTC
0x4003_E000	62	—
0x4003_F000	63	—
0x4004_0000	64	LPTMR
0x4004_1000	65	System register file
0x4004_2000	66	—
0x4004_3000	67	—
0x4004_4000	68	—
0x4004_5000	69	—
0x4004_6000	70	—
0x4004_7000	71	SIM low-power logic
0x4004_8000	72	SIM
0x4004_9000	73	Port A multiplexing control
0x4004_A000	74	Port B multiplexing control
0x4004_B000	75	Port C multiplexing control
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	LPUART0
0x4005_5000	85	LPUART1
0x4005_6000	86	—
0x4005_7000	87	—
0x4005_8000	88	LTC
0x4005_9000	89	RSIM

Table continues on the next page...

Table 5-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4005_A000	90	DCDC
0x4005_B000	91	BLE Link Layer (BTLL)
0x4005_C000	92	XCVR
0x4005_D000	93	
0x4005_E000	94	—
0x4005_F000	95	Generic FSK Link Layer
0x4006_0000	96	—
0x4006_1000	97	—
0x4006_2000	98	CMT
0x4006_3000	99	—
0x4006_4000	100	MCG
0x4006_5000	101	System Oscillator (OSC)
0x4006_6000	102	I^2C 0
0x4006_7000	103	I^2C 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	—
0x4007_3000	115	CMP0
0x4007_4000	116	VREF
0x4007_5000	117	—
0x4007_6000	118	—
0x4007_7000	119	—
0x4007_8000	120	—
0x4007_9000	121	—
0x4007_A000	122	—
0x4007_B000	123	—
0x4007_C000	124	LLWU
0x4007_D000	125	PMC
0x4007_E000	126	SMC
0x4007_F000	127	RCM
0x400F_F000	128	GPIO controller

Table 5-2. Peripheral bridge 0 slot assignments

System 32-bit base address	Slot number	Module

5.6.3 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.

5.7 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 5-3. 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
		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 6
Clock Distribution

6.1 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.

6.2 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.

6.3 High-Level device clocking diagram

The device includes the following clock sources:

- RF Reference oscillator. This supports a 26 MHz or 32 MHz 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
- 32 kHz RTC oscillator. This is used as the clock for the RTC and the deepsleep clock for the Radio Link Layers.

- 32 kHz IRC. This is used as the reference for the FLL at reset, and is a clock option for some peripherals
- 4 MHz 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 32 kHz 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*

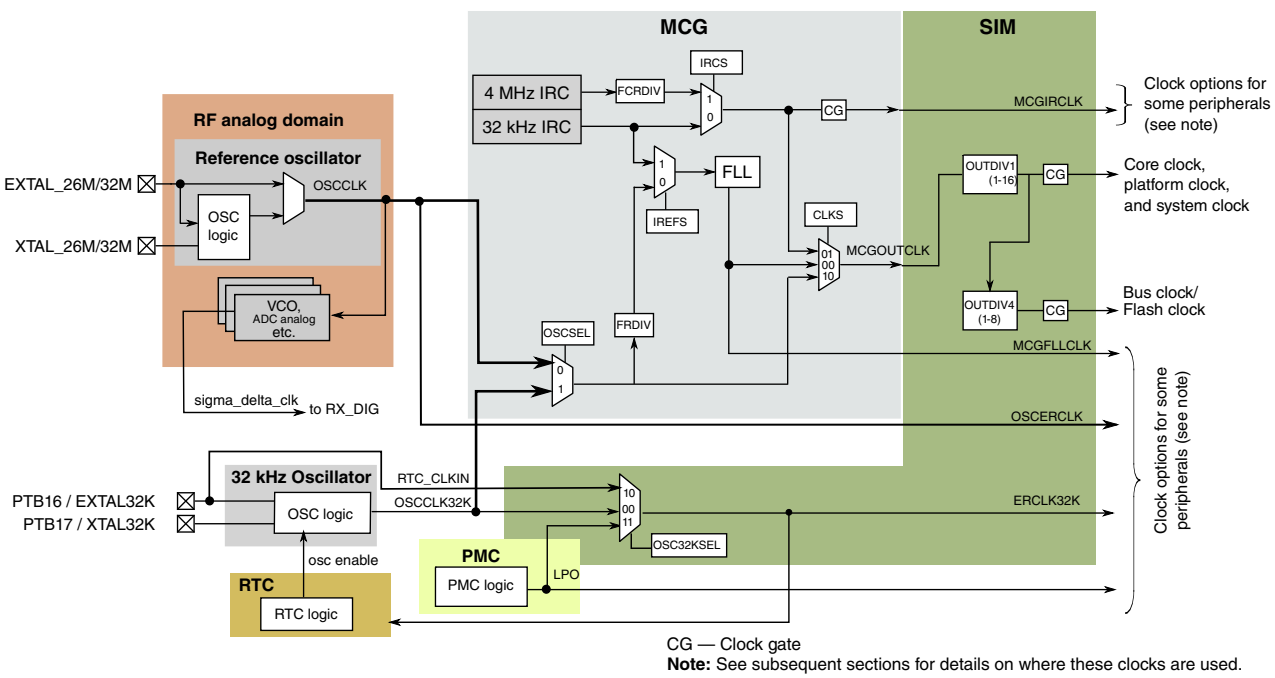


Figure 6-1. Clocking diagram

NOTE

The OSCSEL bit is configured only before enabling the OSCCLK and the OSCCLK32K.

6.4 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 32 kHz oscillator. A 32.768 kHz crystal or clock is used to support RTC features. If RTC is not needed, a 32.768 kHz crystal or clock is needed to support radio deepsleep mode (DSM)
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 Radio; when using Radio, LPO should not be selected.
LPO	PMC 1kHz output

6.4.1 Device clock summary

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

Table 6-1. 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

Table continues on the next page...

Table 6-1. Clock Summary (continued)

Clock name	Run mode clock frequency	VLPR mode clock frequency	Clock source	Clock is disabled when…
MCGFLLCLK	Up to 48 MHz	N/A	MCG	MCG clock controls are not enabled (in all stop modes)
Core clock	Up to 48 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all wait and stop modes
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)	26 MHz or 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_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
LPUART clock	Up to 48 MHz	Up to 4 MHz	MCGIRCLK, MCGFLLCLK, or OSCERCLK	SIM_SOPT2[LPUART0SRC]=00 or selected clock source disabled.

1. BLPE: MCG mode where MCGOUT is derived from an external oscillator. For KW36/35, 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.

6.5 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.

6.5.1 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 FTFE_FOPT register control the reset value of the core clock, system clock, bus clock, and flash clock dividers (in the SIM's CLKDIV1 register) as shown below:

FTFE_FOPT [4,0]	Core/system clock	Bus/Flash clock	Description
00	0x7 (divide by 8), ~2.5MHz	0x1 (divide by 2), ~1.25MHz	Slow clock boot
01	0x3 (divide by 4), ~5MHz	0x1 (divide by 2), ~2.5MHz	Slow clock boot
10	0x1 (divide by 2), ~10MHz	0x1 (divide by 2), ~5MHz	Slow clock 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 FTFE_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.

6.5.2 VLPR mode clocking

For KW36/35 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

6.6 Reference Oscillator Enable Sources

The Reference oscillator is always used by the radio, but can also be used by the MCU. As such, the Reference Oscillator will need to be enabled whenever either subsystem needs the clock. The enable sources for the system oscillator are described as follows.

The radio link layers enable the reference oscillator when they are not in deepsleep mode (DSM).

The oscillator can be enabled manually for Run/Wait modes, and also optionally for Stop mode, by setting the appropriate bits in the RSIM

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

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

The oscillator can also be enabled by the XTAL_OUT_EN pins.

6.7 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 SIM chapter for more information.

6.8 Module clocks

The following table summarizes the clocks associated with each module.

Table 6-2. 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	—	—
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		—

Table continues on the next page...

Table 6-2. Module clocks (continued)

Module	Bus interface clock	Internal clocks	I/O interface clocks
Flash memory	Flash clock	—	—
Analog			
ADC	Bus clock	OSCERCLK	—
CMP	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
LPUART	Bus clock	LPUART clock	—
FlexCAN	Bus clock	OSCERCLK/2	—
Human-machine interfaces			
GPIO	Platform clock	—	—
Security			
AESA	System clock	—	—
TRNG	Bus clock	—	—
Radio			
RF/Analog	—	OSC CLK	—
BTLL	OSCERCLK/2	OSCERCLK, ERCLK32K	—
RSIM	Bus clock	OSC32KCLK	
XCVR_PHY	OSCERCLK	OSCERCLK, sigma_delta_clk	—
Generic FSK	OSCERCLK	OSCERCLK	

1. The 32 kHz osc will be controlled via RTC.

6.8.1 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.

6.8.2 COP clocking

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

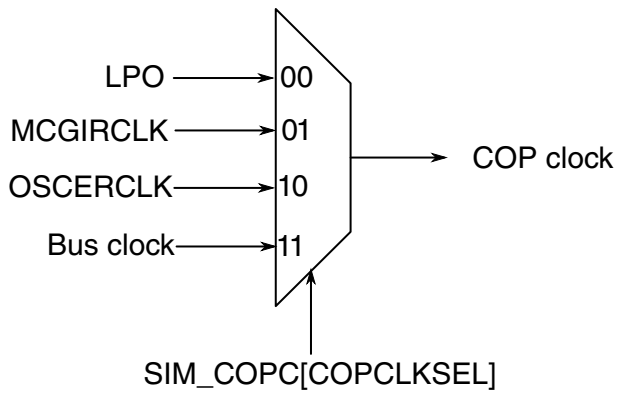


Figure 6-2. COP clock generation

6.8.3 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.

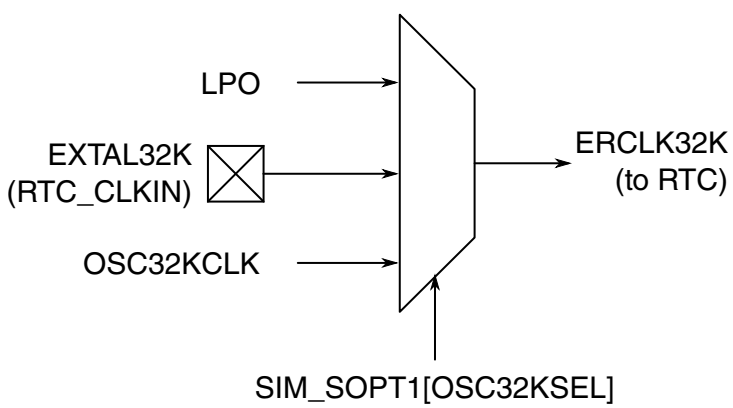


Figure 6-3. RTC clock generation

6.8.4 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.

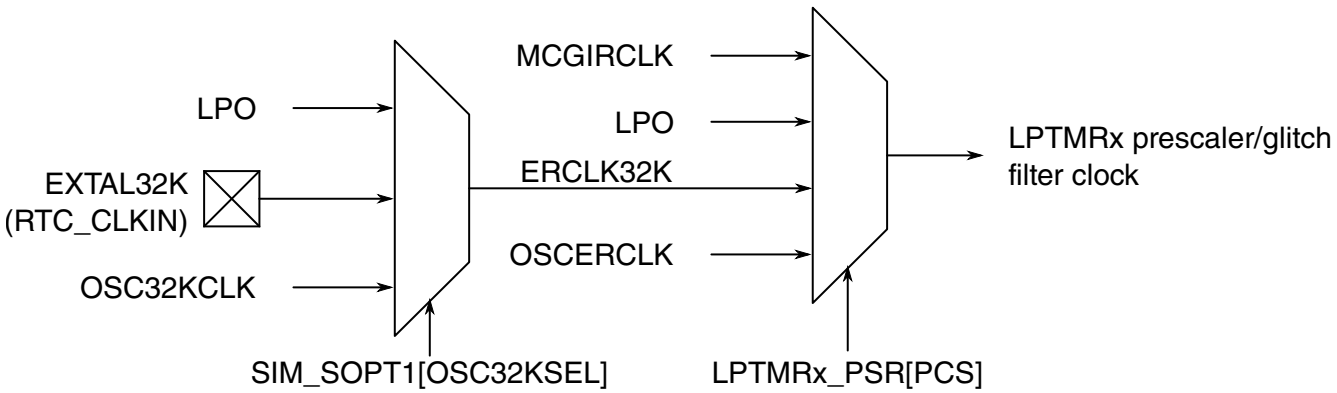


Figure 6-4. LPTMRx prescaler/glitch filter clock generation

6.8.5 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.

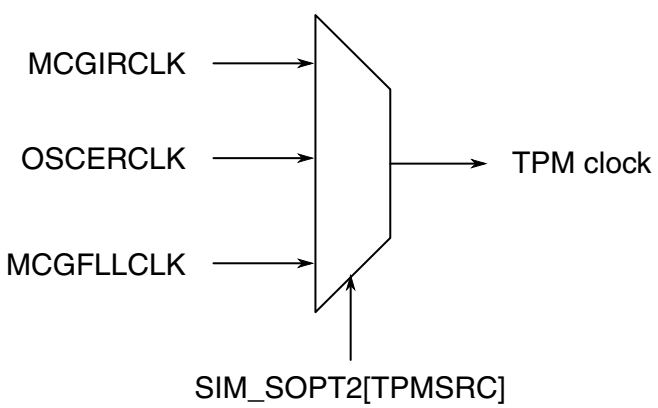


Figure 6-5. TPM clock generation

6.8.6 LPUART clocking

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

NOTE

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

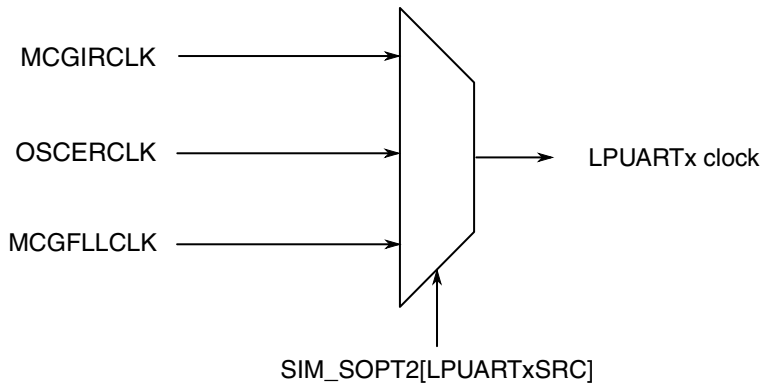


Figure 6-6. LPUART clock generation

6.8.7 FlexCAN clocking

FlexCAN clock can be the bus clock or OSCERCLK.

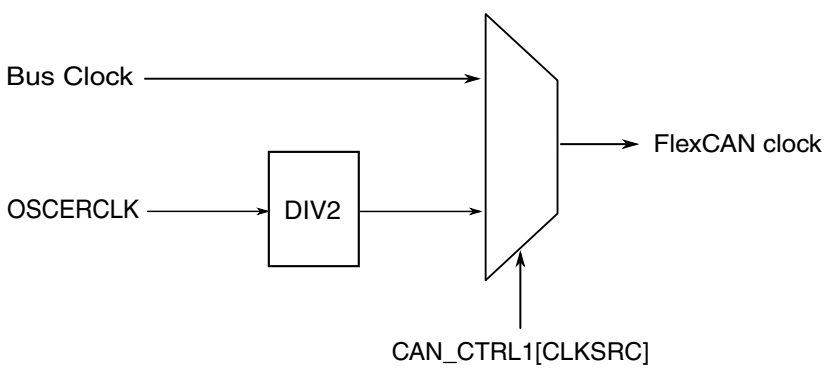


Figure 6-7. FlexCAN Clocking

Chapter 7
Reset and Boot

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

7.1 Introduction

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

Table 7-1. 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.

7.2 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.

7.2.1 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 System Reset Status register, RCM_SRS0, are set following a POR.

7.2.2 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 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)

7.2.2.1 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.

7.2.2.1.1 $\overline{\text{RESET}}$ 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.

7.2.2.2 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.

7.2.2.3 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.

7.2.2.4 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.

7.2.2.5 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.

7.2.2.6 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.

7.2.2.7 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.

7.2.2.8 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.

7.2.2.9 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.

7.2.3 MCU resets

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

7.2.3.1 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.

7.2.3.2 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).

7.2.3.3 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.

7.2.3.4 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.

7.2.3.5 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 RESET pin. It resets the radio digital logic which remains in state-retention during VLLS3 and VLLS2 modes.

7.2.3.6 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).

7.2.3.7 Chip Reset

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

7.2.4 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 FTFE_FOPT[RESET_PIN_CFG] option bit to 0 (See Table 7-2). 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.

7.3 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.

7.3.1 Boot sources

The Cortex-M0+ 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.

7.3.2 FOPT boot options

The Flash Option (FOPT) register in the Flash Memory module (FTFE_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 FTFE_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 FTFE_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 7-2. Flash Option Register (FTFE_FOPT) bit definitions

Bit Num	Field	Value	Definition
7-6	Reserved	Reserved for future expansion.	
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.

Table continues on the next page...

Table 7-2. Flash Option Register (FTFE_FOPT) bit definitions (continued)

Bit Num	Field	Value	Definition
			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.	
		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).

7.3.3 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 MCG is 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 (FTFE_FOPT). If the bits associated with

FTFE_FOPT[LPBOOT] are programmed for an alternate clock divider reset value, the system/core clock is switched to a slower clock speed.

5. When flash Initialization completes, the RESET pin is released. If RESET continues to be asserted (an indication of a slow rise time on the RESET pin or external drive in low), the system continues to be held in reset. Once the 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 NMI input and FTFE_FOPT[NMI_DIS] (See Table 7-2) :
   - If the NMIinput is high or the NMI function is disabled in FTFE_FOPT, the CPU begins execution at the PC location.
   - If the NMI input is low and the NMI function is enabled in FTFE_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 8
Power Management

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

8.1 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 or bypass configuration.

For more details see DCDC chapter.

8.2 Clocking Modes

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

8.2.1 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.

8.2.2 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.

8.2.3 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.

8.2.4 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.

8.2.5 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.

8.3 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 either wait or stop depending on the SLEEPDEEP bit in Cortex-M0+ System Control Register. The primary modes are augmented in a number of ways to provide lower power based on application needs.

Table 8-1. 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 remain 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 DC-DC is configured for continuous mode in VLPR/W	—	Radio operation is possible only when DC-DC 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 DC-DC 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 can be operational. Biasing is disabled when DC-DC is configured for continuous mode in VLPS	Interrupt	
LLS3 (Low Leakage Stop)	State retention power mode. LLWU, LPTMR, RTC, CMP 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 BLE/ Generic FSK DSM[2] logic can be active using the 32 kHz clock
LLS2 (Low Leakage Stop)	State retention power mode. LLWU, LPTMR, RTC, CMP can be operational. 16 KB or 32 KB of programmable RAM can be powered on. All of the radio SOG logic is in state retention	Wakeup Interrupt	

Table continues on the next page...

Table 8-1. Power modes (At 25 deg C) (continued)

Power mode	Description	CPU recovery method	Radio
VLLS3 (Very Low Leakage Stop3)	Full SRAM retention. LLWU, LPTMR, RTC, CMP 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 BLE/Generic FSK DSM logic can be active using the 32 kHz clock
VLLS2 (Very Low Leakage Stop2)	Partial SRAM retention. 16 KB or 32 KB of programmable RAM can be powered on.. LLWU, LPTMR, RTC, CMP can be operational.All of the radio SOG logic is in state retention -	Wakeup Reset	
VLLS1 (Very Low Leakage Stop1) with RTC + 32 kHz 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 can be operational.	Wakeup Reset	
VLLS0 (Very Low Leakage Stop0) with Brown-out Detection	VLLS0 is not supported with DC-DC 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) without Brown-out Detection	VLLS0 is not supported with DC-DC buck 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 Radio's deepsleep mode. DSM does not refer to the ARM sleep deep mode.

8.4 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.

NOTE

When the OSCCLK, the 32 MHz or 26 MHz crystal or external clock from EXTAL/XTAL, which serves as the Radio clock, is used as KW36/35 system clock, the RSIM_CONTROL[STOP_ACK_OVRD] bit should be set to avoid STOP/VLPS/LLSx/VLLSx mode entrance failure. Alternatively, KW36/35 system clock can be switched to run from an internal mode, such as FEI, before entering low power mode.

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 and part of RAM can be retained in VLLS2 if RAM2PO is enabled.

8.5 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 8-2. 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, VLLS2, OFF in VLLS0/1
LVD	disabled	disabled	ON	disabled	disabled	disabled
Brown-out Detection	ON	ON	ON	ON	ON	ON in VLLS1/2/3, optionally disabled in VLLS0[3]
DMA	FF Async operation in CPO	FF	Async operation	Async operation	static	OFF
Watchdog	FF static in CPO	FF	Optional with clock source enabled in stop mode FF in PSTOP2	Optional with clock source enabled in stop mode	static	OFF
Clocks						
1 kHz LPO	ON	ON	ON	ON	ON	ON in VLLS1/2/3, OFF in VLLS0
Reference oscillator (OSC)[4]	Should be OFF, except when DCDC configured for continuous mode	Should be OFF, except when DCDC configured for continuous mode	ON as needed	Should be OFF, except when DCDC configured for continuous mode	OFF	OFF
32 kHz 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	1 MHz max	OFF	OFF	OFF	OFF

Table continues on the next page...

Table 8-2. Module operation in low power modes (continued)

Modules	VLPR	VLPW	Stop	VLPS	LLSx	VLLSx
	OFF in CPO		24 MHz max in PSTOP2 from RUN 1 MHz max in PSTOP2 from VLPR			
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						
UART0/1 (LPUART)	1 Mbps Async operation in CPO	1 Mbps	Async operation FF in PSTOP2	Async operation	static	OFF
DSPI1/0	500 kbps	500 kbps	FF in PSTOP2	static	static	OFF
I^2C0	50 kbps static, address match wakeup in CPO	50 kbps	static, address match wakeup FF in PSTOP2	static, address match wakeup	static	OFF
I^2C1	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
FlexCAN	FF wake-up in CPO	FF	Wake-up FF in PSTOP2	Wake-up	static	static
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]

Table continues on the next page...

Table 8-2. Module operation in low power modes (continued)

Modules	VLPR	VLPW	Stop	VLPS	LLSx	VLLSx
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
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
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)	DCDC configured for continuous mode : FF Otherwise: static (must be in DSM)	FF	DCDC configured for continuous mode : FF Otherwise: static (must be in DSM)	static (state retention) DSM timer can be active	VLLS3/2: static (state retention) DSM timer can be active VLLS1/0: OFF
Generic FSK	DCDC configured for continuous mode : FF Otherwise: static (must be in DSM)	DCDC configured for continuous mode : FF Otherwise: static (must be in DSM)	FF	DCDC configured for continuous mode : FF Otherwise: static (must be in DSM)	static (state retention) DSM timer can be active	VLLS3/2: static (state retention) DSM timer can be active VLLS1/0: OFF
PHY_DIG	PHY_DIG operation follows the same behavior as described in the rows above for BTLL.					
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	DCDC configured for continuous mode : Can be enabled by Radio TSM	Can be enabled by Radio TSM	DCDC configured for continuous mode : Can be enabled by Radio TSM	OFF	OFF

Table 8-2. Module operation in low power modes

Modules	VLPR	VLPW	Stop	VLPS	LLSx	VLLSx
	Otherwise: should be OFF	Otherwise: should be OFF		Otherwise: should be 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 32 KHz oscillator can be bypassed, in which case the 32 kHz EXTAL32K 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 32 kHz 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.

Chapter 9
Security

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

9.1 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.

9.2 Flash security

The flash module provides security information to the MCU based on the state held by FTFE_FSEC[SEC]. The MCU, in turn, confirms the security request and limits access to flash resources. During reset, the flash module initializes FTFE_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 FTFE_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 (FTFE_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.

9.3 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.

9.3.1 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.

9.3.2 Firmware Distribution Protection

KW36/35 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 FTFE for flash protection information

Chapter 10
Debug

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

10.1 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)

10.2 Debug port pin descriptions

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

Table 10-1. 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.

10.3 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 10-2. 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

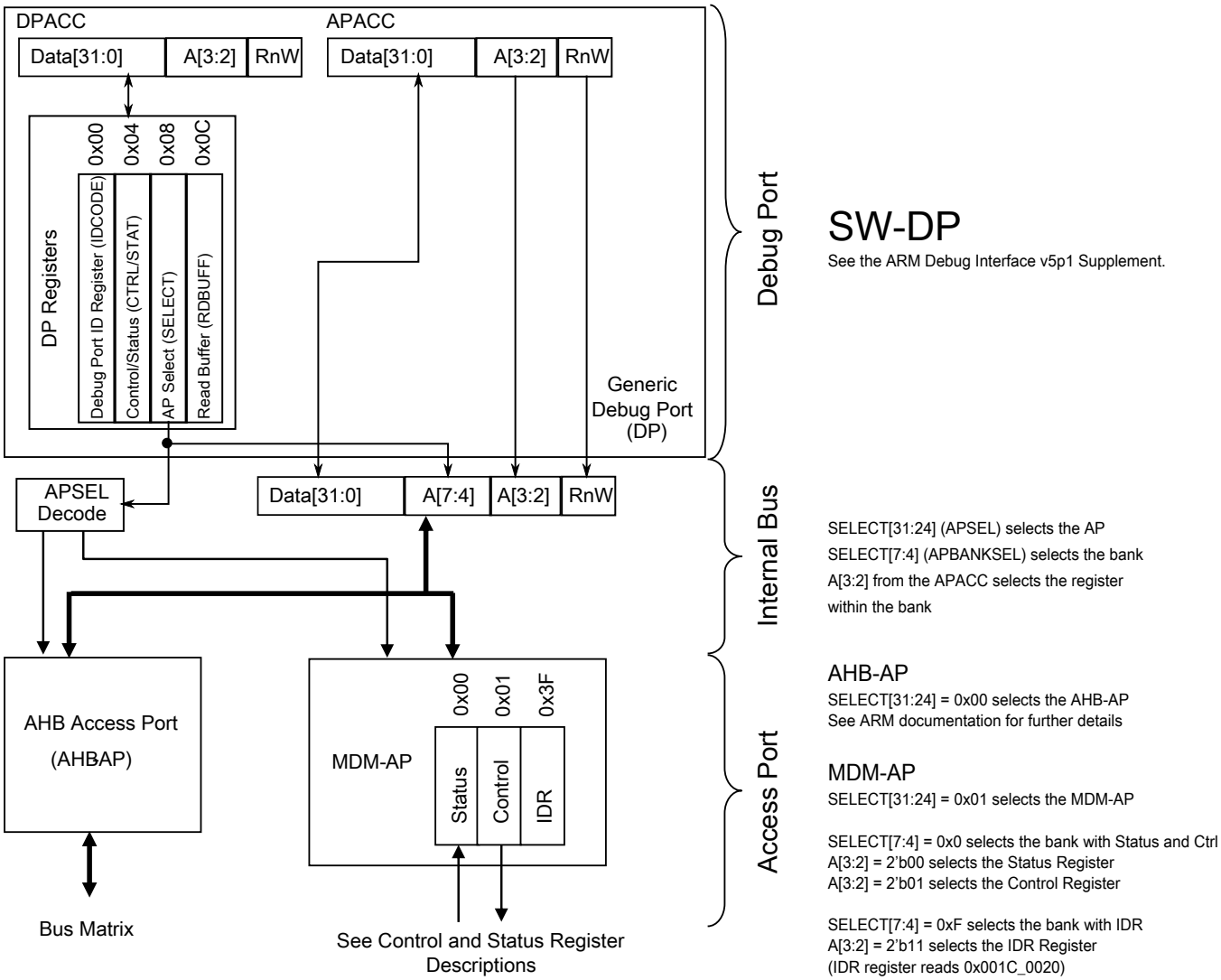


Figure 10-1. MDM AP addressing

10.3.1 MDM-AP Control Register

Table 10-3. 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 10-3. 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 if mass erase is enabled in the MEEN bits and 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.

Table continues on the next page...

Table 10-3. MDM-AP Control register assignments (continued)

Bit	Name	Secure[1]	Description
			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.
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

10.3.2 MDM-AP Status Register

Table 10-4. 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.

Table continues on the next page...

Table 10-4. MDM-AP Status register assignments (continued)

Bit	Name	Description
		0 Low Power Stop Mode is not enabled. 1 Low Power Stop Mode is enabled. 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
18	Core SLEEPING	SLEEPING==1 and SLEEPDEEP==0 indicates wait or VLPW mode. SLEEPING==1 and SLEEPDEEP==1 indicates stop or VLPS mode.
19 – 31	Reserved for future use	Always read 0.

10.4 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.

10.5 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.

10.6 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.

10.7 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 11
Kinetis Flashloader

11.1 Chip-Specific Information

This device has various peripherals (LPUART, I2C, SPI, CAN) supported by the Kinetis Flashloader. The following table shows the pads used by the Kinetis Flashloader.

Table 11-1. Kinetis Flashloader Peripheral Pinmux

Peripheral	Instance	Alt mode	Port
LPUART	0	4	PTC2, LPUART0_RX PTC18, LPUART0_TX
I2C	0	3	PTB0, I2C0_SCL PTB1, I2C0_SDA
SPI	0	2	PTC16, SPI0_SCK PTC17, SPI0_SOUT PTC18, SPI0_SIN PTC19, SPI0_PCS0
CAN	0	9	PTC3, CAN0_TX PTC4, CAN0_RX

11.2 Introduction

The Kinetis devices *that do not have an on-chip ROM* are shipped with the pre-programmed Kinetis Flashloader in the on-chip flash memory, for one-time, in-system factory programming. The Kinetis Flashloader's main task is to load a customer firmware image into the flash memory. The image on the flash has 2 programs: flashloader_loader and flashloader. After a device reset, the flashloader_loader program starts its execution

first. The flashloader_loader program copies the contents of flashloader image from the flash to the on-chip RAM; the device then switches execution to the flashloader program to execute from RAM.

For this device, the Kinetis Flashloader can interface with LPUART, CAN, I2C, and SPI peripherals in slave mode and respond to the commands sent by a master (or host) communicating on one of those ports. The host/master can be a firmware-download application running on a PC or an embedded host communicating with the Kinetis Flashloader. Regardless of the host/master (PC or embedded host), the Kinetis Flashloader always uses a command protocol to communicate with that host/master. Commands are provided to write to memory (flash or RAM), erase flash, and get/set flashloader options and property values. The host application can query the set of available commands.

This chapter describes Kinetis Flashloader features, functionality, command structure and which peripherals are supported.

Features supported by the Kinetis Flashloader :

- Supports LPUART, CAN, I2C, and SPI peripheral interfaces
- Automatic detection of the active peripheral
- LPUARTand CAN peripherals with autobaud
- Common packet-based protocol for all peripherals
- Packet error detection and retransmission
- Protection of RAM used by the flashloader while it is running
- Provides command to read properties of the device, such as flash and RAM size

Table 11-2. Commands supported by the Kinetis Flashloader

Command	Description	When flash security is enabled, then this command is
Call	Runs user application code and returns control to bootloader	Not supported
Execute	Run user application code that never returns control to the flashloader	Not supported
FillMemory	Fill a range of bytes in flash with a word pattern	Not supported
FlashEraseAll	Erase the entire flash array	Not supported
FlashEraseRegion	Erase a range of sectors in flash	Not supported
FlashProgramOnce	Writes data provided in a command packet to a specified range of bytes in the program once field	Not supported
FlashReadOnce	Returns the contents of the program once field by given index and byte count	Not supported
FlashReadResource	Returns the contents of the IFR field or Flash Version ID, by given offset, byte count and option	Not supported
WriteMemory	Write data to memory	Not supported

Table continues on the next page...

Table 11-2. Commands supported by the Kinetis Flashloader (continued)

Command	Description	When flash security is enabled, then this command is
ReadMemory	Read data from memory	Not supported
GetProperty	Get the current value of a property	Supported
Reset	Reset the chip	Supported
SetProperty	Attempt to modify a writable property	Supported
FlashEraseAllUnsecure	Erase the entire flash array, including protected sectors	Supported

11.3 Functional Description

The following sub-sections describe the Kinetis Flashloader functionality.

11.3.1 Memory Maps

While executing, the Kinetis Flashloader uses RAM memory.

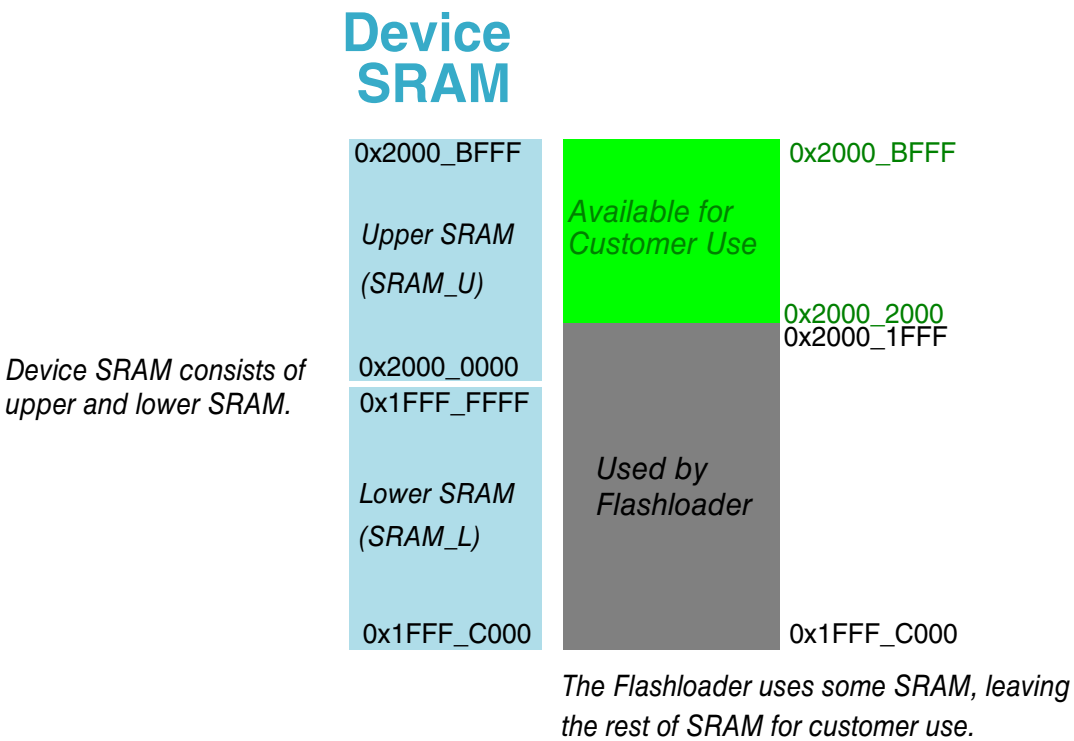


Figure 11-1. Kinetis Flashloader RAM Memory Map

NOTE

The Kinetis Flashloader requires a minimum memory space of 32 KB of RAM. For Kinetis devices with less than this amount of on-chip RAM, the Kinetis Flashloader is not available.

11.3.2 Start-up Process

As the Kinetis Flashloader begins executing, flashloader operations begin:

1. The flashloader's temporary working area in RAM is initialized.
2. All supported peripherals are initialized.
3. The flashloader waits for communication to begin on a peripheral.
   - There is no timeout for the active peripheral detection process.
   - If communication is detected, then all inactive peripherals are shut down, and the command phase is entered.

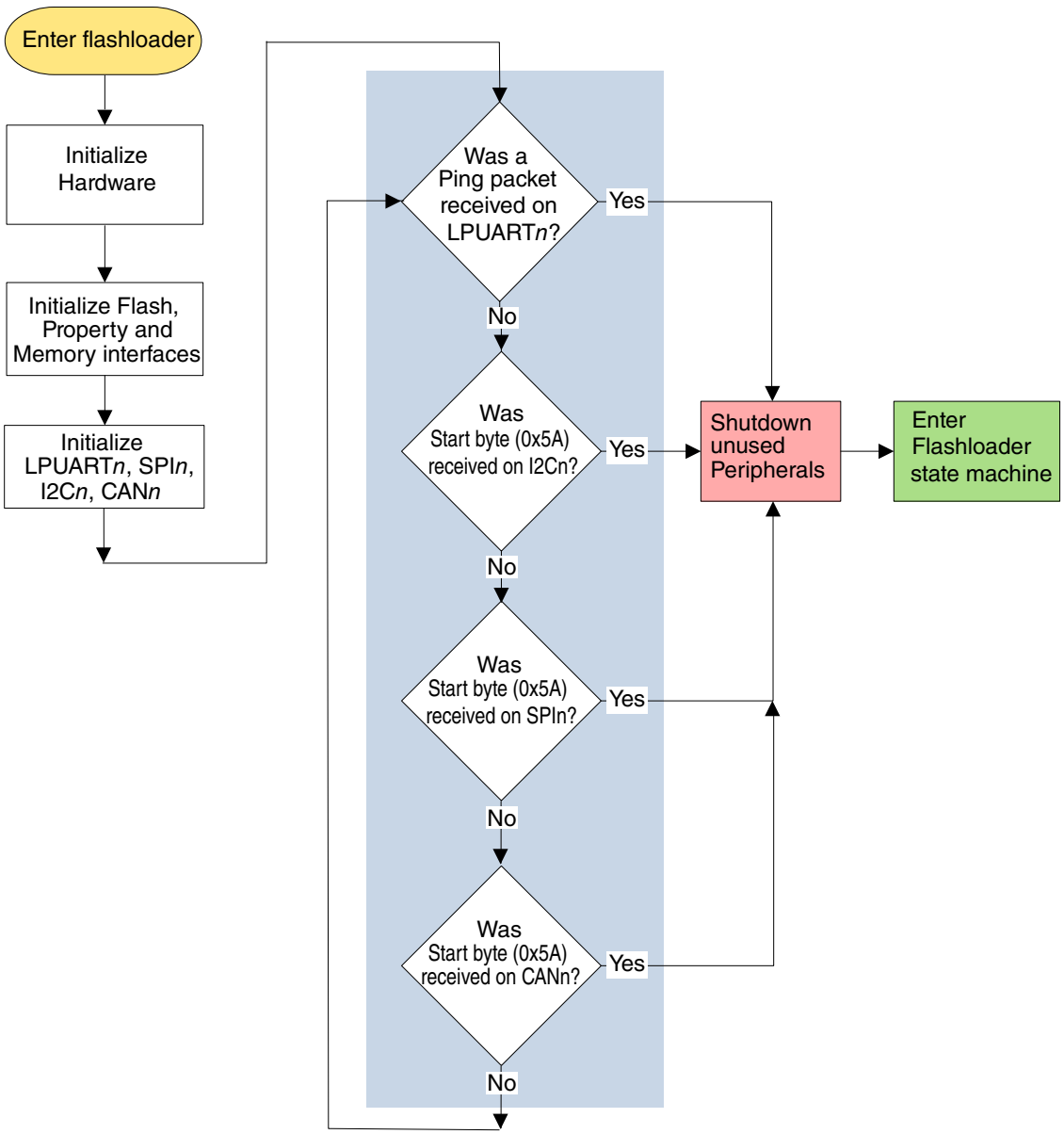


Figure 11-2. Kinetis Flashloader Start-up Flowchart

11.3.3 Clock Configuration

The core runs on the default reset clock (20.9 MHz). The Kinetis Flashloader does not modify any clocks, and after a reset, the core uses the clock configuration of the chip.

11.3.4 Flashloader Protocol

This section explains the general protocol for the packet transfers between the host and the Kinetis Flashloader. The description includes the transfer of packets for different transactions, such as commands with no data phase and commands with incoming or outgoing data phase. The next section describes various packet types used in a transaction.

Each command sent from the host is replied to with a response command.

Commands may include an optional data phase:

- If the data phase is **incoming** (from host to flashloader), then the data phase is part of the **original command**.
- If the data phase is **outgoing** (from flashloader to host), then the data phase is part of the **response command**.

NOTE

In all protocols (described in the next subsections), the Ack sent in response to a Command or Data packet can arrive at any time *before, during, or after* the Command/Data packet has processed.

11.3.4.1 Command with no data phase

The protocol for a command with no data phase contains:

- Command packet (from host)
- Generic response command packet (to host)

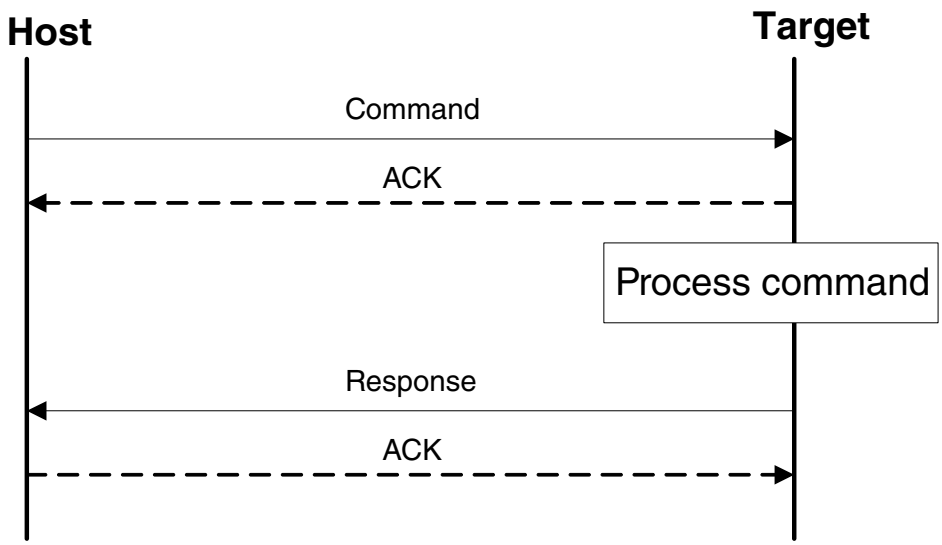


Figure 11-3. Command with No Data Phase

11.3.4.2 Command with incoming data phase

The protocol for a command with an incoming data phase contains:

- Command packet (from host)
- Generic response command packet (to host)
- Incoming data packets (from host)
- Generic response command packet (to host)

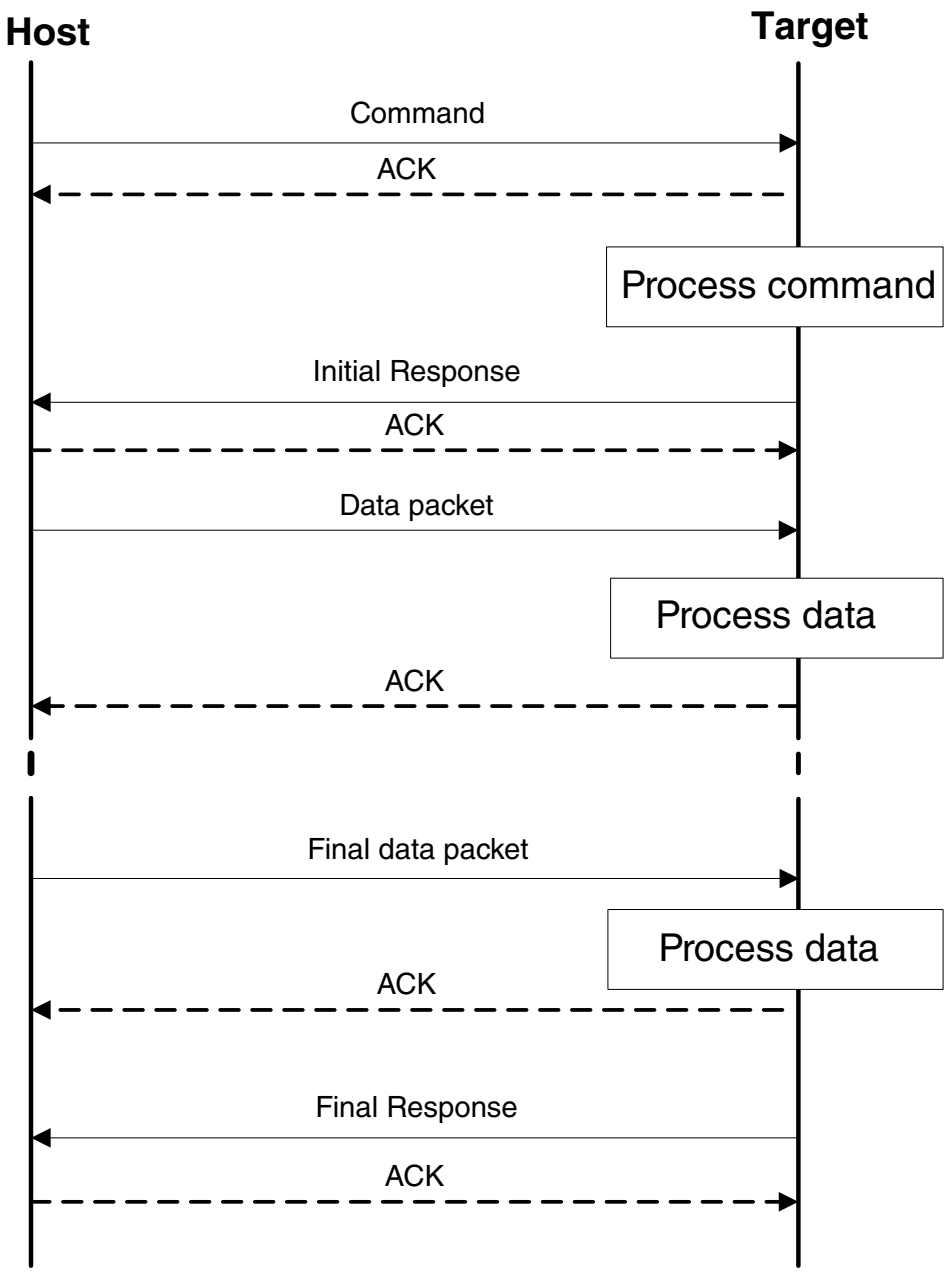


Figure 11-4. Command with incoming data phase

NOTE

- The host may not send any further packets while it (the host) is waiting for the response to a command.
- If the Generic Response packet prior to the start of the data phase does not have a status of kStatus_Success, then the data phase is aborted.
- Data phases may be aborted by the receiving side by sending the final Generic Response early with a status of

kStatus_AbortDataPhase. The host may abort the data phase early by sending a zero-length data packet.
- The final Generic Response packet *sent after the data phase* includes the status for the entire operation.

11.3.4.3 Command with outgoing data phase

The protocol for a command with an outgoing data phase contains:

- Command packet (from host)
- ReadMemory Response command packet (to host) (kCommandFlag_HasDataPhase set)
- Outgoing data packets (to host)
- Generic response command packet (to host)

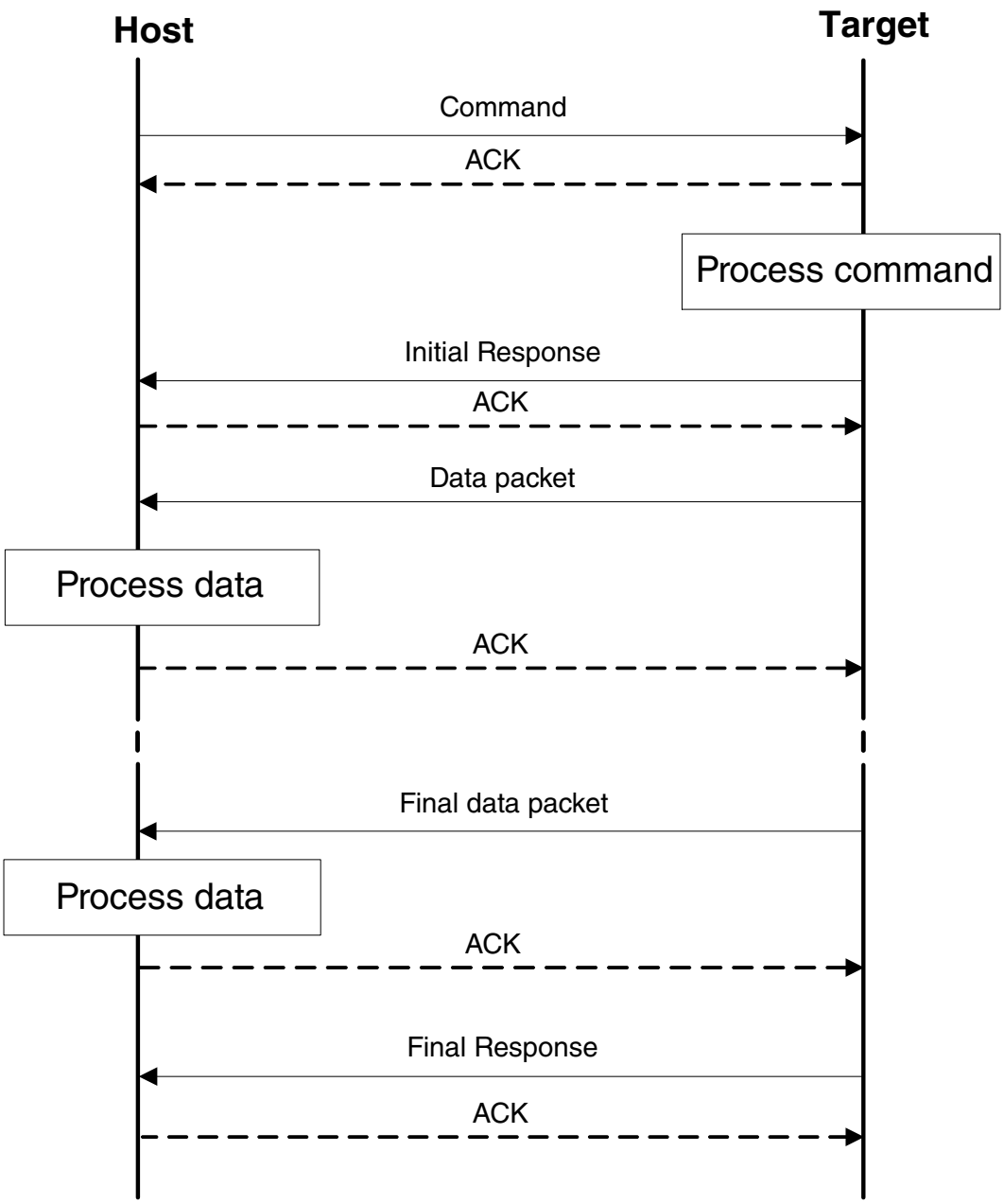


Figure 11-5. Command with outgoing data phase

NOTE

- For the outgoing data phase sequence above, the data phase is really considered part of the response command.
- The host may not send any further packets while it (the host) is waiting for the response to a command.
- If the ReadMemory Response command packet prior to the start of the data phase does not contain the kCommandFlag_HasDataPhase flag, then the data phase is aborted.

- Data phases may be aborted by the host sending the final Generic Response early with a status of kStatus_AbortDataPhase. The sending side may abort the data phase early by sending a zero-length data packet.
- The final Generic Response packet *sent after the data phase* includes the status for the entire operation.

11.3.5 Flashloader Packet Types

The Kinetis Flashloader device works in slave mode. All data communication is initiated by a host, which is either a PC or an embedded host (note that QuadSPI is treated like a storage device, like flash). The Kinetis Flashloader device is the target, which receives a command or data packet. All data communication between host and target is packetized.

NOTE

The term "target" refers to the "Kinetis Flashloader device."

There are 6 types of packets used in the device:

- Ping packet
- Ping Response packet
- Framing packet
- Command packet
- Data packet
- Response packet

All fields in the packets are in little-endian byte order.

11.3.5.1 Ping packet

The Ping packet is the first packet sent from a host to the target (Kinetis Flashloader), to establish a connection on a selected peripheral. For a LPUART peripheral, the Ping packet is used to determine the baudrate. A Ping packet must be sent before any other communications. In response to a Ping packet, the target sends a Ping Response packet.

Table 11-3. Ping Packet Format

Byte #	Value	Name
0	0x5A	start byte
1	0xA6	ping

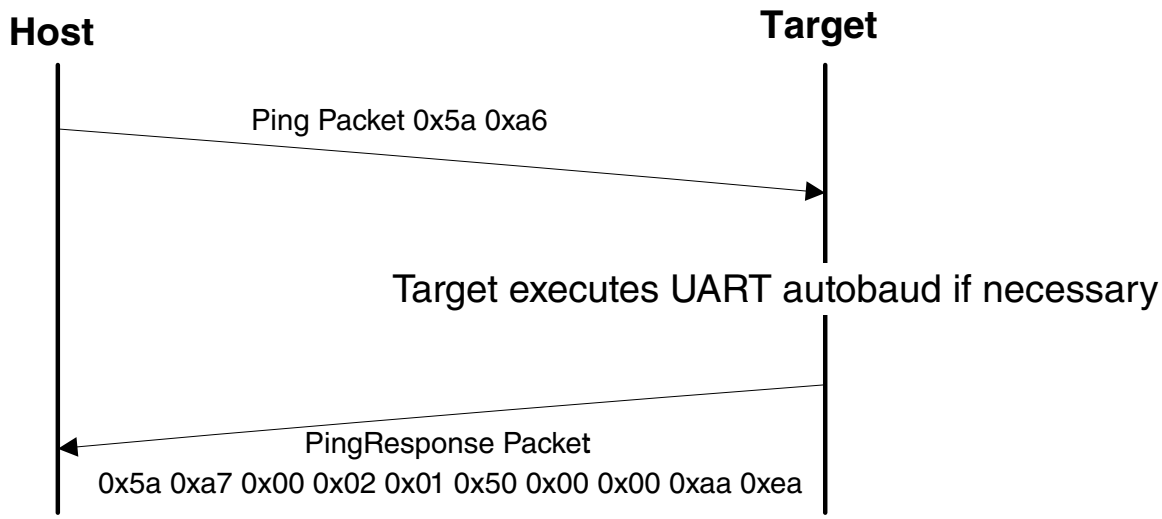


Figure 11-6. Ping Packet Protocol Sequence

11.3.5.2 Ping Response Packet

The target (Kinetis Flashloader) sends a Ping Response packet back to the host after receiving a Ping packet. If communication is over a LPUART peripheral, the target uses the incoming Ping packet to determine the baud rate before replying with the Ping Response packet. Once the Ping Response packet is received by the host, the connection is established, and the host starts sending commands to the target (Kinetis Flashloader).

Table 11-4. Ping Response Packet Format

Byte #	Value	Parameter
0	0x5A	start byte
1	0xA7	Ping response code
2		Protocol bugfix
3		Protocol minor
4		Protocol major
5		Protocol name = 'P' (0x50)
6		Options low
7		Options high
8		CRC16 low
9		CRC16 high

11.3.5.3 Framing Packet

The framing packet is used for flow control and error detection, and it (the framing packet) wraps command and data packets as well.

The framing packet described in this section is used for serial peripherals including LPUART, I2C, SPI, and CAN.

Table 11-5. Framing Packet Format

Byte #	Value	Parameter	
0	0x5A	start byte	
1		packetType	
2		length_low	Length is a 16-bit field that specifies the entire command or data packet size in bytes.
3		length_high	
4		crc16_low	This is a 16-bit field. The CRC16 value covers entire framing packet, including the start byte and command or data packets, but does not include the CRC bytes. See the CRC16 algorithm after this table.
5		crc16_high	
6 . . .n		Command or Data packet payload	

A special framing packet that contains only a start byte and a packet type is used for synchronization between the host and target.

Table 11-6. Special Framing Packet Format

Byte #	Value	Parameter
0	0x5A	start byte
1	0xA*n*	packetType

The Packet Type field specifies the type of the packet from one of the defined types (below):

Table 11-7. packetType Field

packetType	Name	Description
0xA1	kFramingPacketType_Ack	The previous packet was received successfully; the sending of more packets is allowed.
0xA2	kFramingPacketType_Nak	The previous packet was corrupted and must be re-sent.
0xA3	kFramingPacketType_AckAbort	Data phase is being aborted.
0xA4	kFramingPacketType_Command	The framing packet contains a command packet payload.
0xA5	kFramingPacketType_Data	The framing packet contains a data packet payload.
0xA6	kFramingPacketType_Ping	Sent to verify the other side is alive. Also used for UART autobaud.

Table continues on the next page...

Table 11-7. packetType Field (continued)

packetType	Name	Description
0xA7	kFramingPacketType_PingResponse	A response to Ping; contains the framing protocol version number and options.

This device uses the Cyclic Redundancy Check module (CRC) to perform the CRC algorithm. See the CRC chapter for more details.

11.3.5.4 Command packet

The command packet carries a 32-bit command header and a list of 32-bit parameters.

Table 11-8. Command Packet Format

Command Packet Format (32 bytes)										
Command Header (4 bytes)				28 bytes for Parameters (Max 7 parameters)						
Tag	Flags	Rsvd	Param Count	Param1 (32-bit)	Param2 (32-bit)	Param3 (32-bit)	Param4 (32-bit)	Param5 (32-bit)	Param6 (32-bit)	Param7 (32-bit)
byte 0	byte 1	byte 2	byte 3							

Table 11-9. Command Header Format

Byte #	Command Header Field	
0	Command or Response tag	The command header is 4 bytes long, with these fields.
1	Flags	
2	Reserved. Should be 0x00.	
3	ParameterCount	

The header is followed by 32-bit parameters up to the value of the ParameterCount field specified in the header. Because a command packet is 32 bytes long, only 7 parameters can fit into the command packet.

Command packets are also used by the target to send responses back to the host. As mentioned earlier, command packets and data packets are embedded into framing packets for all of the transfers.

Table 11-10. Commands that are supported

Command	Name
0x01	FlashEraseAll
0x02	FlashEraseRegion
0x03	ReadMemory

Table continues on the next page...

Table 11-10. Commands that are supported (continued)

Command	Name
0x04	WriteMemory
0x05	FillMemory
0x06	Reserved
0x07	GetProperty
0x08	Reserved
0x09	Execute
0x0A	Call
0x0B	Reset
0x0C	SetProperty
0x0D	FlashEraseAllUnsecure
0x0E	FlashProgramOnce
0x0F	FlashReadOnce
0x10	FlashReadResource
0x11	Reserved
0x12	Reserved

Table 11-11. Responses that are supported

Response	Name
0xA0	GenericResponse
0xA3	ReadMemoryResponse (used for sending responses to ReadMemory command only)
0xA7	GetPropertyResponse (used for sending responses to GetProperty command only)
0xAF	FlashReadOnceResponse (used for sending responses to FlashReadOnce command only)
0xB0	FlashReadResourceResponse (used for sending responses to FlashReadResource command only)

Flags: Each command packet contains a Flag byte. Only bit 0 of the flag byte is used. If bit 0 of the flag byte is set to 1, then data packets will follow in the command sequence. The number of bytes that will be transferred in the data phase is determined by a command-specific parameter in the parameters array.

ParameterCount: The number of parameters included in the command packet.

Parameters: The parameters are word-length (32 bits). With the default maximum packet size of 32 bytes, a command packet can contain up to 7 parameters.

11.3.5.5 Data packet

The data packet carries just the data, either host sending data to target, or target sending data to host. The data transfer direction is determined by the last command sent from the host. The data packet is also wrapped within a framing packet, to ensure the correct packet data is received.

The contents of a data packet are simply the data itself. There are no other fields, so that the most data per packet can be transferred. Framing packets are responsible for ensuring that the correct packet data is received.

11.3.5.6 Response packet

The responses are carried using the same command packet format wrapped with framing packet data. Types of responses include:

- GenericResponse
- GetPropertyResponse
- ReadMemoryResponse
- FlashReadOnceResponse
- FlashReadResourceResponse

GenericResponse: After the Kinetis Flashloader has processed a command, the flashloader will send a generic response with status and command tag information to the host. The generic response is the last packet in the command protocol sequence. The generic response packet contains the framing packet data and the command packet data (with generic response tag = 0xA0) and a list of parameters (defined in the next section). The parameter count field in the header is always set to 2, for status code and command tag parameters.

Table 11-12. GenericResponse Parameters

Byte #	Parameter	Descripton
0 - 3	Status code	The Status codes are errors encountered during the execution of a command by the target (Kinetis Flashloader). If a command succeeds, then a kStatus_Success code is returned. Table 11-50, Kinetis Flashloader Status Error Codes, lists the status codes returned to the host by the Kinetis Flashloader.
4 - 7	Command tag	The Command tag parameter identifies the response to the command sent by the host.

GetPropertyResponse: The GetPropertyResponse packet is sent by the target in response to the host query that uses the GetProperty command. The GetPropertyResponse packet contains the framing packet data and the command packet data, with the command/response tag set to a GetPropertyResponse tag value (0xA7).

The parameter count field in the header is set to greater than 1, to always include the status code and one or many property values.

Table 11-13. GetPropertyResponse Parameters

Byte #	Value	Parameter
0 - 3		Status code
4 - 7		Property value
. . .		. . .
		Can be up to maximum 6 property values, limited to the size of the 32-bit command packet and property type.

ReadMemoryResponse: The ReadMemoryResponse packet is sent by the target in response to the host sending a ReadMemory command. The ReadMemoryResponse packet contains the framing packet data and the command packet data, with the command/response tag set to a ReadMemoryResponse tag value (0xA3), the flags field set to kCommandFlag_HasDataPhase (1).

The parameter count set to 2 for the status code and the data byte count parameters shown below.

Table 11-14. ReadMemoryResponse Parameters

Byte #	Parameter	Descripton
0 - 3	Status code	The status of the associated Read Memory command.
4 - 7	Data byte count	The number of bytes sent in the data phase.

FlashReadOnceResponse: The FlashReadOnceResponse packet is sent by the target in response to the host sending a FlashReadOnce command. The FlashReadOnceResponse packet contains the framing packet data and the command packet data, with the command/response tag set to a FlashReadOnceResponse tag value (0xAF), and the flags field set to 0. The parameter count is set to 2 plus *the number of words* requested to be read in the FlashReadOnceCommand.

Table 11-15. FlashReadOnceResponse Parameters

Byte #	Value	Parameter
0 – 3		Status Code

Table continues on the next page...

Table 11-15. FlashReadOnceResponse Parameters (continued)

4 – 7		Byte count to read
…		…
		Can be up to 20 bytes of requested read data.

The FlashReadResourceResponse packet is sent by the target in response to the host sending a FlashReadResource command. The FlashReadResourceResponse packet contains the framing packet data and command packet data, with the command/response tag set to a FlashReadResourceResponse tag value (0xB0), and the flags field set to kCommandFlag_HasDataPhase (1).

Table 11-16. FlashReadResourceResponse Parameters

Byte #	Value	Parameter
0 – 3		Status Code
4 – 7		Data byte count

11.3.6 Flashloader Command API

All Kinetis Flashloader command APIs follow the command packet format that is wrapped by the framing packet, as explained in previous sections.

- For a list of commands supported by the Flashloader, see Table 11-2, Commands supported.
- For a list of status codes returned by the Kinetis Flashloader, see Table 11-50, Kinetis Flashloader Status Error Codes.

NOTE

All the examples in this section depict byte traffic on serial peripherals that use framing packets.

11.3.6.1 Call command

The Call command will execute a function that is written in memory at the address sent in the command. The address needs to be a valid memory location residing in accessible flash (internal or external) or in RAM. The command supports the passing of one 32-bit argument. Although the command supports a stack address, at this time the call will still take place using the current stack pointer. After execution of the function, a 32-bit return value will be returned in the generic response message.

Table 11-17. Parameters for Call Command

Byte #	Command
0 - 3	Call address
4 - 7	Argument word
8 - 11	Stack pointer

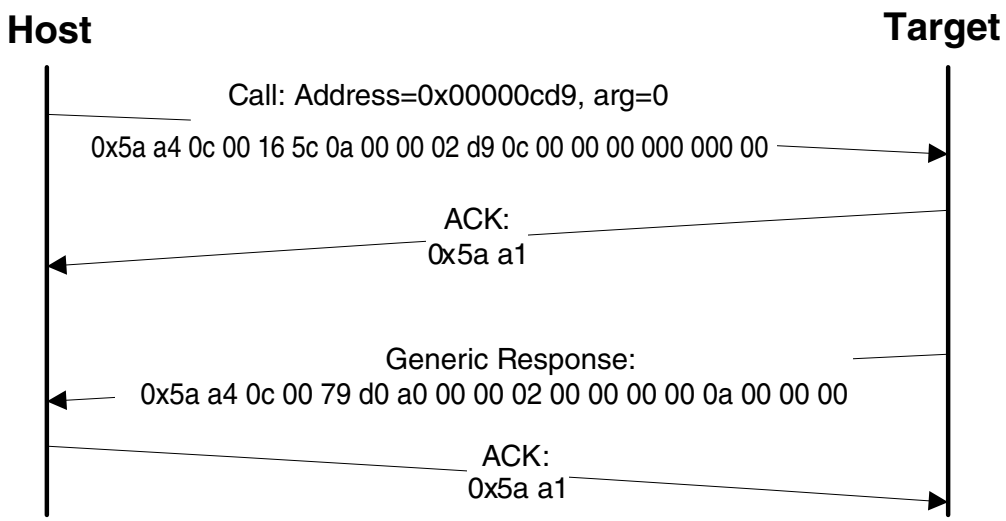


Figure 11-7. Protocol Sequence for Call Command

Response: The target (Kinetis Flashloader) will return a GenericResponse packet with a status code either set to the return value of the function called or set to kStatus_InvalidArgument (105).

11.3.6.2 GetProperty command

The GetProperty command is used to query the flashloader about various properties and settings. Each supported property has a unique 32-bit tag associated with it. The tag occupies the first parameter of the command packet. The target returns a GetPropertyResponse packet with the property values for the property identified with the tag in the GetProperty command.

Properties are the defined units of data that can be accessed with the GetProperty or SetProperty commands. Properties may be read-only or read-write. All read-write properties are 32-bit integers, so they can easily be carried in a command parameter.

For a list of properties and their associated 32-bit property tags supported by the Kinetis Flashloader, see Table 11-45.

The 32-bit property tag is the only parameter required for GetProperty command.

Table 11-18. Parameters for GetProperty Command

Byte #	Command
0 - 3	Property tag

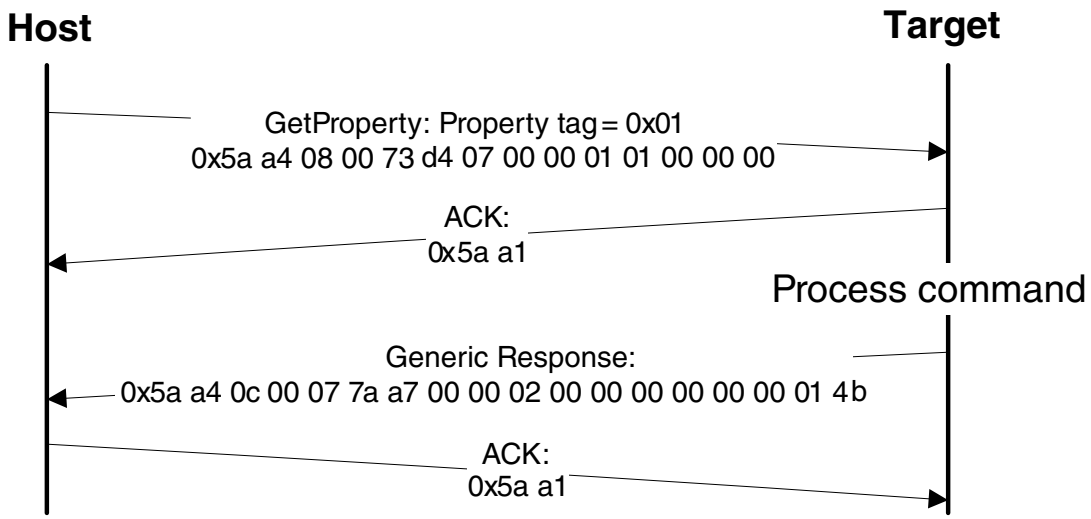


Figure 11-8. Protocol Sequence for GetProperty Command

Table 11-19. GetProperty Command Packet Format (Example)

GetProperty	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x08 0x00
	crc16	0x73 0xD4
Command packet	commandTag	0x07 – GetProperty
	flags	0x00
	reserved	0x00
	parameterCount	0x01
	propertyTag	0x00000001 - CurrentVersion

The GetProperty command has no data phase.

Response: In response to a GetProperty command, the target will send a GetPropertyResponse packet with the response tag set to 0xA7. The parameter count indicates the number of parameters sent for the property values, with the first parameter showing status code 0, followed by the property value(s). The next table shows an example of a GetPropertyResponse packet.

Table 11-20. GetProperty Response Packet Format (Example)

GetPropertyResponse	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x0c 0x00 (12 bytes)
	crc16	0x07 0x7a
Command packet	responseTag	0xA7
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	status	0x00000000
	propertyValue	0x0000014b - CurrentVersion

11.3.6.3 SetProperty command

The SetProperty command is used to change or alter the values of the properties or options in the Kinetis Flashloader. However, the SetProperty command can only change the value of properties that are writable—see Table 11-45, Properties used by Get/SetProperty Commands. If you try to set a value for a read-only property, then the Kinetis Flashloader will return an error.

The property tag and the new value to set are the 2 parameters required for the SetProperty command.

Table 11-21. Parameters for SetProperty Command

Byte #	Command
0 - 3	Property tag
4 - 7	Property value

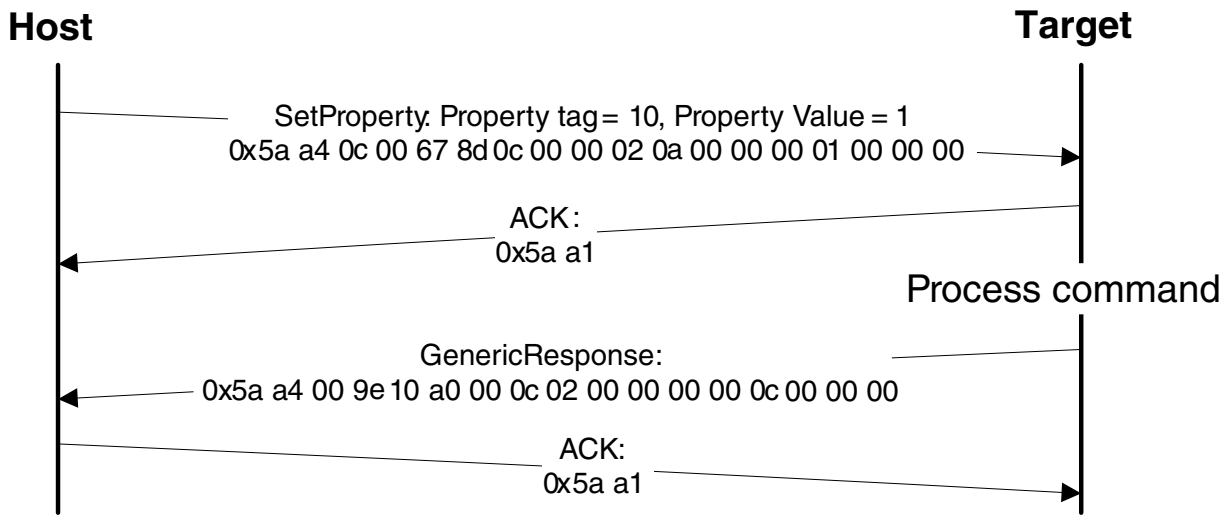


Figure 11-9. Protocol Sequence for SetProperty Command

Table 11-22. SetProperty Command Packet Format (Example)

SetProperty	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x0C 0x00
	crc16	0x67 0x8D
Command packet	commandTag	0x0C – SetProperty with property tag 10
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	propertyTag	0x0000000A - VerifyWrites
	propertyValue	0x00000001

The SetProperty command has no data phase.

Response: The target (Kinetis Flashloader) will return a GenericResponse packet with one of following status codes:

Table 11-23. SetProperty Response Status Codes

Status Code
kStatus_Success
kStatus_ReadOnly
kStatus_UnknownProperty
kStatus_InvalidArgument

11.3.6.4 FlashEraseAll command

The FlashEraseAll command performs an erase of the entire flash memory. If any flash regions are protected, then the FlashEraseAll command will fail and return an error status code. Executing the FlashEraseAll command will release flash security if it (flash security) was enabled, by setting the FTFA_FSEC register. However, the FSEC field of the flash configuration field is erased, so unless it is reprogrammed, the flash security will be re-enabled after the next system reset. The Command tag for FlashEraseAll command is 0x01 set in the commandTag field of the command packet.

The FlashEraseAll command requires 1 parameter: memoryId.

Table 11-24. Parameters for FlashEraseAll command

Bytes	Parameter
0 - 3	MemoryId • 0x00 - Internal PFlash • 0x01 - QuadSPI memory

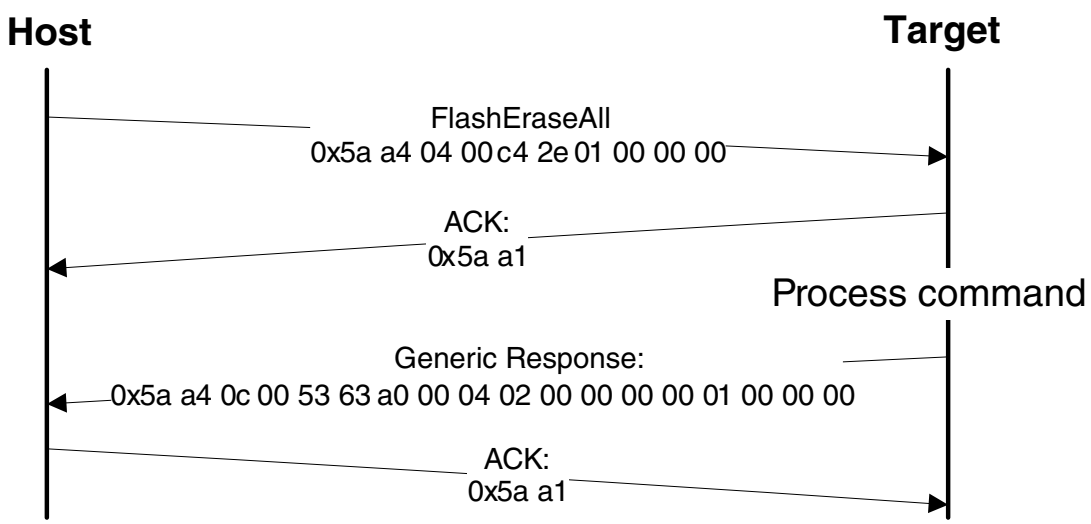


Figure 11-10. Protocol Sequence for FlashEraseAll Command

Table 11-25. FlashEraseAll Command Packet Format (Example)

FlashEraseAll	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x04 0x00
	crc16	0xC4 0x2E

Table continues on the next page...

Table 11-25. FlashEraseAll Command Packet Format (Example) (continued)

FlashEraseAll	Parameter	Value
Command packet	commandTag	0x01 - FlashEraseAll
	flags	0x00
	reserved	0x00
	parameterCount	0x00
	MemoryID	• If MemoryID = 0x00h, then internal flash. • If MemoryID = 0x01h, then QSPI0 memory.

The FlashEraseAll command has no data phase.

Response: The target (Kinetis Flashloader) will return a GenericResponse packet with status code either set to kStatus_Success for successful execution of the command, or set to an appropriate error status code.

11.3.6.5 FlashEraseRegion command

The FlashEraseRegion command performs an erase of one or more sectors of the flash memory or a specified range of flash within the connected SPI flash devices.

The start address and number of bytes are the 2 parameters required for the FlashEraseRegion command. The start and byte count parameters must be , or the FlashEraseRegion command will fail and return kStatus_FlashAlignmentError (0x101). If the region specified does not fit in the flash memory space, the FlashEraseRegion command will fail and return kStatus_FlashAddressError (0x102). If any part of the region specified is protected, the FlashEraseRegion command will fail and return kStatus_MemoryRangeInvalid (0x10200).

Table 11-26. Parameters for FlashEraseRegion Command

Byte #	Parameter
0 - 3	Start address
4 - 7	Byte count

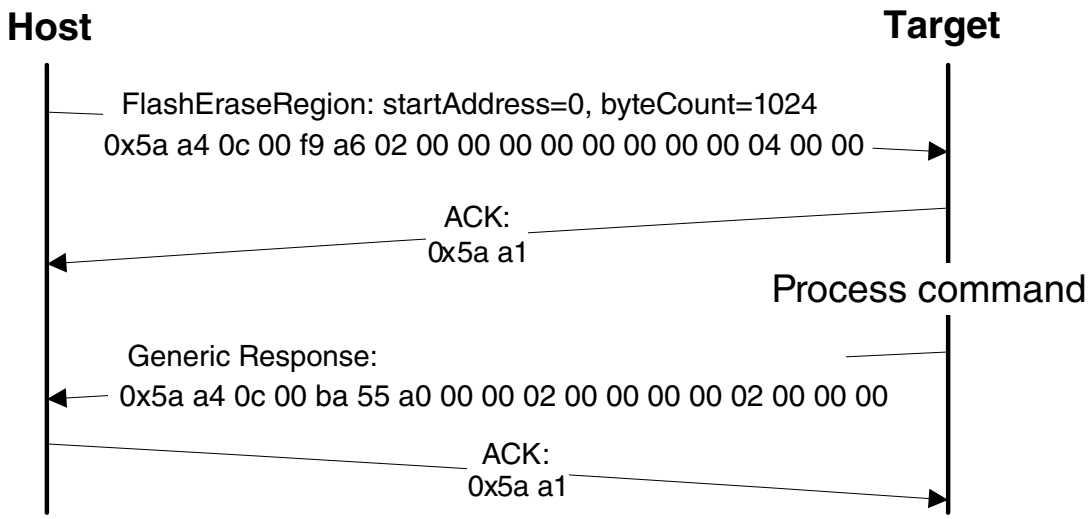


Figure 11-11. Protocol Sequence for FlashEraseRegion Command

Table 11-27. FlashEraseRegion Command Packet Format (Example)

FlashEraseRegion	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x0C 0x00
	crc16	0xF9 0x A6
Command packet	commandTag	0x02, kCommandTag_FlashEraseRegion
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	startAddress	0x00 0x00 0x00 0x00 (0x0000_0000)
	byte count	0x00 0x04 0x00 0x00 (0x400)

The FlashEraseRegion command has no data phase.

Response: The target (Kinetis Flashloader) will return a GenericResponse packet with one of following error status codes.

Table 11-28. FlashEraseRegion Response Status Codes

Status Code
kStatus_Success (0x0)
kStatus_MemoryRangeInvalid (0x10200)
kStatus_FlashAlignmentError (0x101)
kStatus_FlashAddressError (0x102)
kStatus_FlashAccessError (0x103)
kStatus_FlashProtectionViolation (0x104)
kStatus_FlashCommandFailure (0x105)

11.3.6.6 FlashEraseAllUnsecure command

The FlashEraseAllUnsecure command performs a mass erase of the flash memory, including protected sectors. Flash security is immediately disabled if it (flash security) was enabled, and the FSEC byte in the flash configuration field at address 0x40C is programmed to 0xFE. However, if the mass erase enable option in the FSEC field is disabled, then the FlashEraseAllUnsecure command will fail.

The FlashEraseAllUnsecure command requires no parameters.

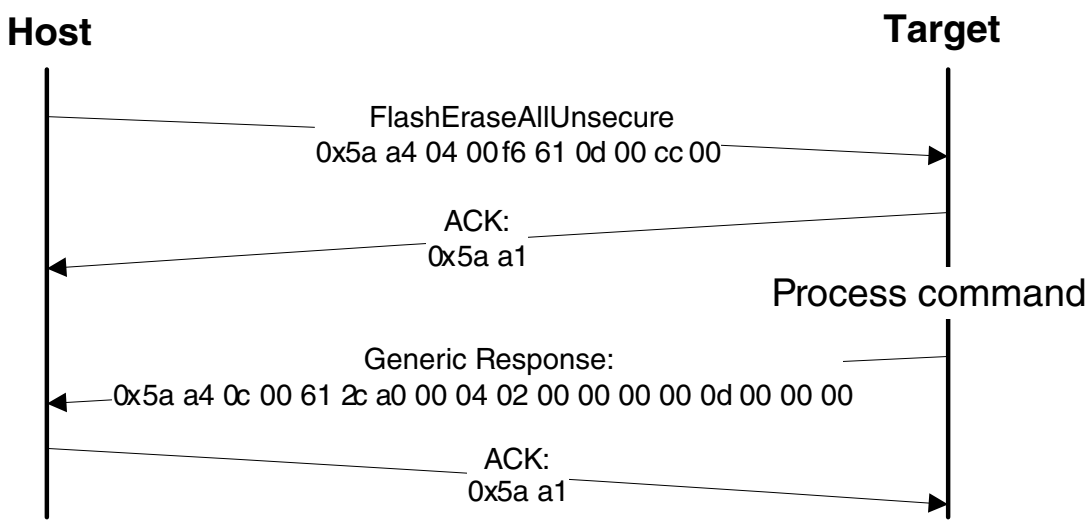


Figure 11-12. Protocol Sequence for FlashEraseAll Command

Table 11-29. FlashEraseAllUnsecure Command Packet Format (Example)

FlashEraseAllUnsecure	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x04 0x00
	crc16	0xF6 0x61
Command packet	commandTag	0x0D - FlashEraseAllUnsecure
	flags	0x00
	reserved	0x00
	parameterCount	0x00

The FlashEraseAllUnsecure command has no data phase.

Response: The target (Kinetis Flashloader) will return a GenericResponse packet with status code either set to kStatus_Success for successful execution of the command, or set to an appropriate error status code.

11.3.6.7 FillMemory command

The FillMemory command fills a range of bytes in memory with a data pattern. It follows the same rules as the WriteMemory command. The difference between FillMemory and WriteMemory is that a data pattern is included in FillMemory command parameter, and there is no data phase for the FillMemory command, while WriteMemory does have a data phase.

Table 11-30. Parameters for FillMemory Command

Byte #	Command
0 - 3	Start address of memory to fill
4 - 7	Number of bytes to write with the pattern • The start address should be 32-bit aligned. • The number of bytes must be evenly divisible by 4.
8 - 11	32-bit pattern

- To fill with a byte pattern (8-bit), the byte must be replicated 4 times in the 32-bit pattern.
- To fill with a short pattern (16-bit), the short value must be replicated 2 times in the 32-bit pattern.

For example, to fill a byte value with 0xFE, the word pattern would be 0xFEFEFEFE; to fill a short value 0x5AFE, the word pattern would be 0x5AFE5AFE.

Special care must be taken when writing to flash.

- First, any flash sector written to must have been previously erased with a FlashEraseAll or FlashEraseRegion command.
- Writing to flash requires the start address to be 8-byte aligned ([2:0] = 000) .
- If the VerifyWrites property is set to true, then writes to flash will also perform a flash verify program operation.

When writing to RAM, the start address need not be aligned, and the data will not be padded.

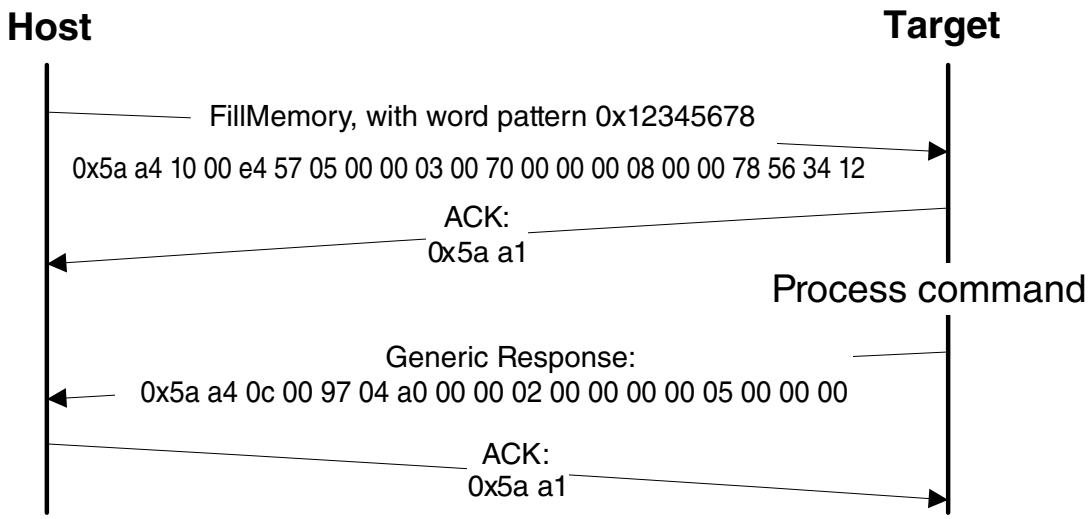


Figure 11-13. Protocol Sequence for FillMemory Command

Table 11-31. FillMemory Command Packet Format (Example)

FillMemory	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x10 0x00
	crc16	0xE4 0x57
Command packet	commandTag	0x05 – FillMemory
	flags	0x00
	Reserved	0x00
	parameterCount	0x03
	startAddress	0x00007000
	byteCount	0x00000800
	patternWord	0x12345678

The FillMemory command has no data phase.

Response: upon successful execution of the command, the target (Kinetis Flashloader) will return a GenericResponse packet with a status code set to kStatus_Success, or to an appropriate error status code.

11.3.6.8 FlashProgramOnce command

The FlashProgramOnce command writes data (that is provided in a command packet) to a specified range of bytes in the program once field. Special care must be taken when writing to the program once field.

- The program once field only supports programming once, so any attempted to reprogram a program once field will get an error response.
- Writing to the program once field requires the byte count to be 4-byte aligned or 8-byte aligned.

The FlashProgramOnce command uses 3 parameters: index, byteCount, data.

Table 11-32. Parameters for FlashProgramOnce Command

Byte #	Command
0 - 3	Index of program once field
4 - 7	Byte count (must be evenly divisible by 4)
8 - 11	Data
12 - 16	Data

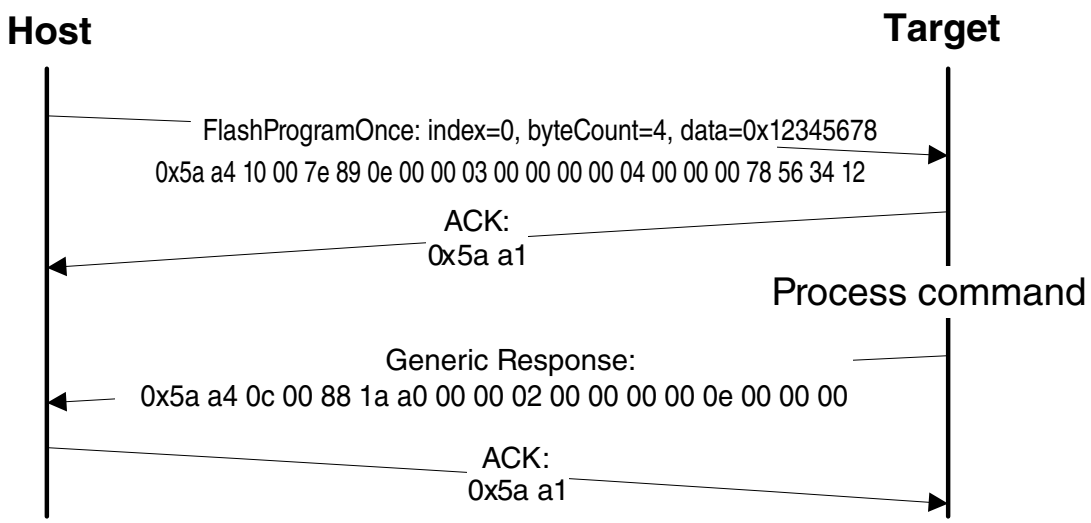


Figure 11-14. Protocol Sequence for FlashProgramOnce Command

Table 11-33. FlashProgramOnce Command Packet Format (Example)

FlashProgramOnce	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x10 0x00
	crc16	0x7E4 0x89
Command packet	commandTag	0x0E – FlashProgramOnce
	flags	0
	reserved	0
	parameterCount	3
	index	0x0000_0000

Table continues on the next page...

Table 11-33. FlashProgramOnce Command Packet Format (Example) (continued)

FlashProgramOnce	Parameter	Value
	byteCount	0x0000_0004
	data	0x1234_5678

Response: upon successful execution of the command, the target (Kinetis Flashloader) will return a GenericResponse packet with a status code set to kStatus_Success, or to an appropriate error status code.

11.3.6.9 FlashReadOnce command

The FlashReadOnce command returns the contents of the program once field by given index and byte count. The FlashReadOnce command uses 2 parameters: index and byteCount.

Table 11-34. Parameters for FlashReadOnce Command

Byte #	Parameter	Description
0 - 3	index	Index of the program once field (to read from)
4 - 7	byteCount	Number of bytes to read and return to the caller

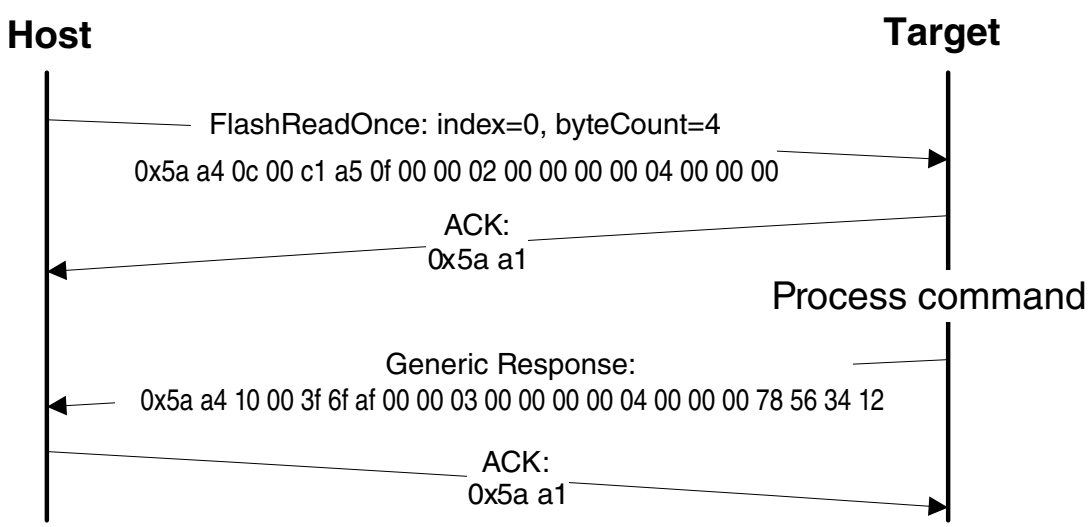


Figure 11-15. Protocol Sequence for FlashReadOnce Command

Table 11-35. FlashReadOnce Command Packet Format (Example)

FlashReadOnce	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4
	length	0x0C 0x00
	crc	0xC1 0xA5
Command packet	commandTag	0x0F – FlashReadOnce
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	index	0x0000_0000
	byteCount	0x0000_0004

Table 11-36. FlashReadOnce Response Format (Example)

FlashReadOnce Response	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4
	length	0x10 0x00
	crc	0x3F 0x6F
Command packet	commandTag	0xAF
	flags	0x00
	reserved	0x00
	parameterCount	0x03
	status	0x0000_0000
	byteCount	0x0000_0004
	data	0x1234_5678

Response: upon successful execution of the command, the target (Kinetis Flashloader) will return a FlashReadOnceResponse packet with a status code set to kStatus_Success, a byte count and corresponding data read from Program Once Field upon successful execution of the command, or will return with a status code set to an appropriate error status code and a byte count set to 0.

11.3.6.10 FlashReadResource command

The FlashReadResource command returns the contents of the IFR field or Flash Version ID, by given offset, byte count, and option. The FlashReadResource command uses 3 parameters: start address, byteCount, option.

Table 11-37. Parameters for FlashReadResource Command

Byte #	Parameter	Command
0 - 3	start address	Start address of specific non-volatile memory to be read
4 - 7	byteCount	Byte count to be read
8 - 11	option	0: IFR 1: Flash Version ID

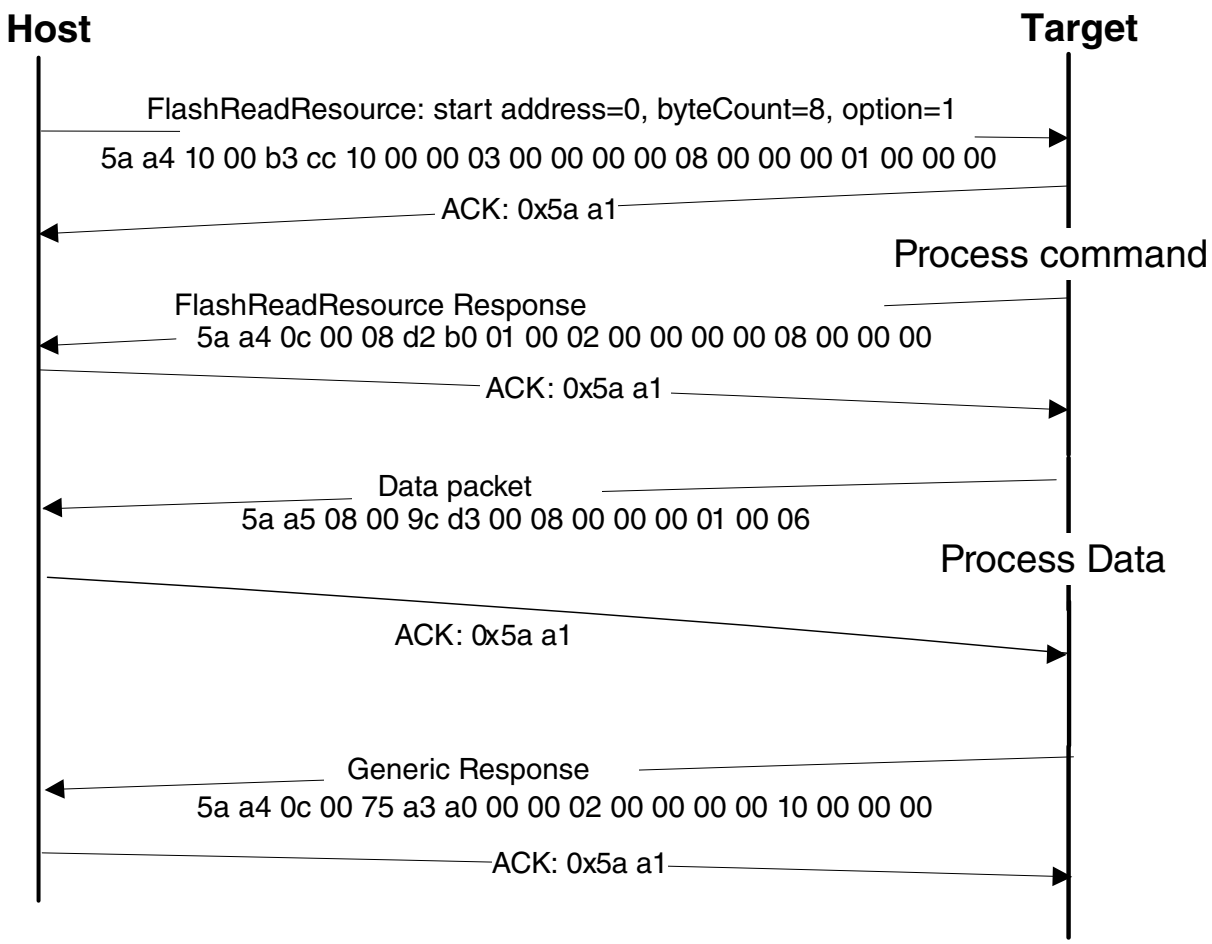


Figure 11-16. Protocol Sequence for FlashReadResource Command

Table 11-38. FlashReadResource Command Packet Format (Example)

FlashReadResource	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4
	length	0x10 0x00
	crc	0xB3 0xCC
Command packet	commandTag	0x10 – FlashReadResource
	flags	0x00

Table continues on the next page...

Table 11-38. FlashReadResource Command Packet Format (Example) (continued)

FlashReadResource	Parameter	Value
	reserved	0x00
	parameterCount	0x03
	startAddress	0x0000_0000
	byteCount	0x0000_0008
	option	0x0000_0001

Table 11-39. FlashReadResource Response Format (Example)

FlashReadResource Response	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4
	length	0x0C 0x00
	crc	0xD2 0xB0
Command packet	commandTag	0xB0
	flags	0x01
	reserved	0x00
	parameterCount	0x02
	status	0x0000_0000
	byteCount	0x0000_0008

Data phase: The FlashReadResource command has a data phase. Because the target (Kinetis Flashloader) works in slave mode, the host must pull data packets until the number of bytes of data *specified in the byteCount parameter of FlashReadResource command* are received by the host.

11.3.6.11 WriteMemory command

The WriteMemory command writes data provided in the data phase to a specified range of bytes in memory (flash or RAM). However, if flash protection is enabled, then writes to protected sectors will fail.

Special care must be taken when writing to flash.

- First, any flash sector written to must have been previously erased with a FlashEraseAll or FlashEraseRegion command.
- Writing to flash requires the start address to be 8-byte aligned ([2:0] = 000).
- If the VerifyWrites property is set to true, then writes to flash will also perform a flash verify program operation.

When writing to RAM, the start address need not be aligned, and the data will not be padded.

The start address and number of bytes are the 2 parameters required for WriteMemory command.

Table 11-40. Parameters for WriteMemory Command

Byte #	Command
0 - 3	Start address
4 - 7	Byte count

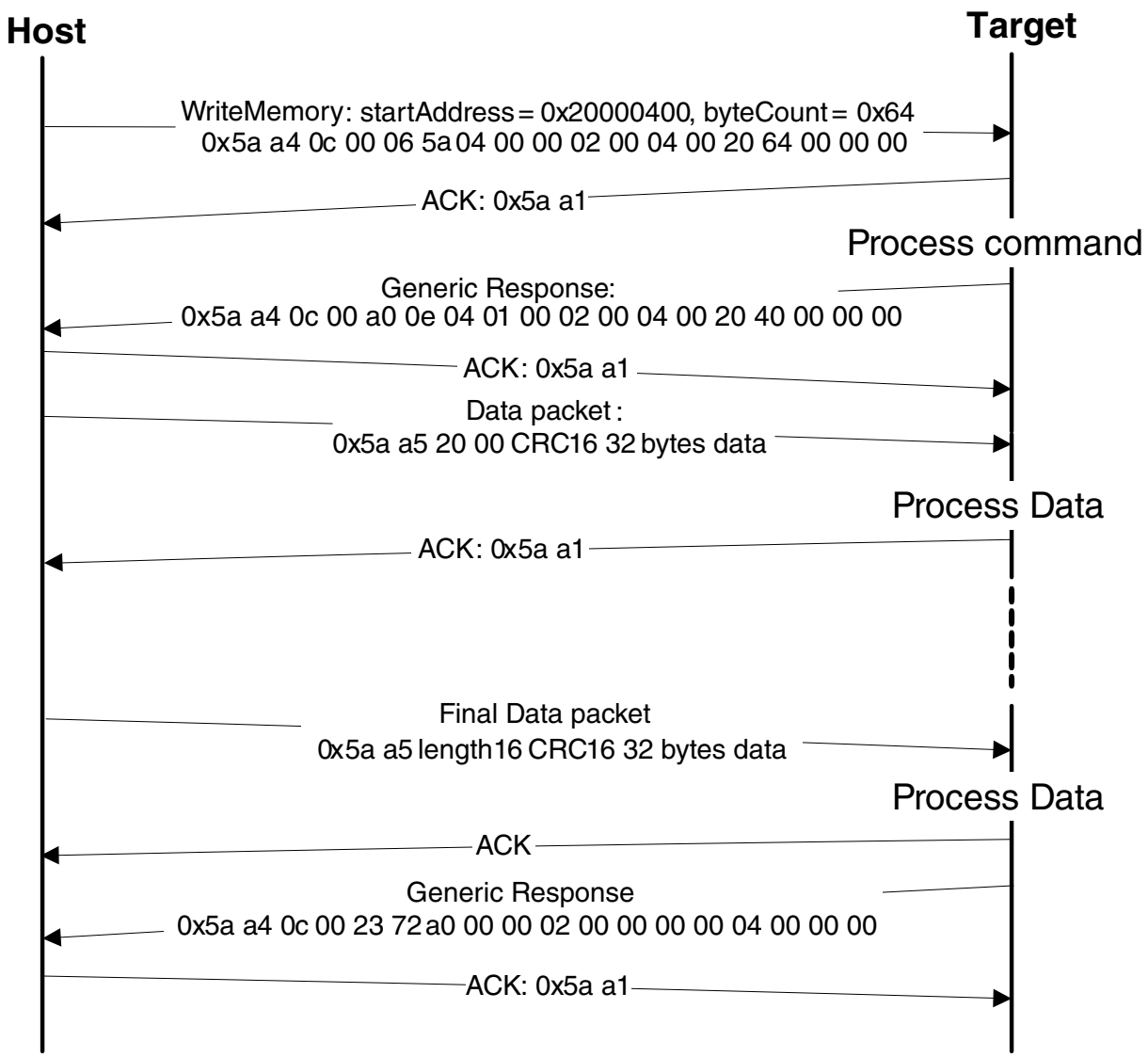


Figure 11-17. Protocol Sequence for WriteMemory Command

Table 11-41. WriteMemory Command Packet Format (Example)

WriteMemory	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x0C 0x00
	crc16	0x06 0x5A
Command packet	commandTag	0x04 - writeMemory
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	startAddress	0x20000400
	byteCount	0x00000064

Data Phase: The WriteMemory command has a data phase; the host will send data packets until the number of bytes of data specified in the byteCount parameter of the WriteMemory command are received by the target.

Response: The target (Kinetis Flashloader) will return a GenericResponse packet with a status code set to kStatus_Success upon successful execution of the command, or to an appropriate error status code.

11.3.6.12 Read memory command

The ReadMemory command returns the contents of memory at the given address, for a specified number of bytes. This command can read any region of memory accessible by the CPU and not protected by security.

The start address and number of bytes are the 2 parameters required for ReadMemory command.

Table 11-42. Parameters for read memory command

Byte	Parameter	Description
0-3	Start address	Start address of memory to read from
4-7	Byte count	Number of bytes to read and return to caller

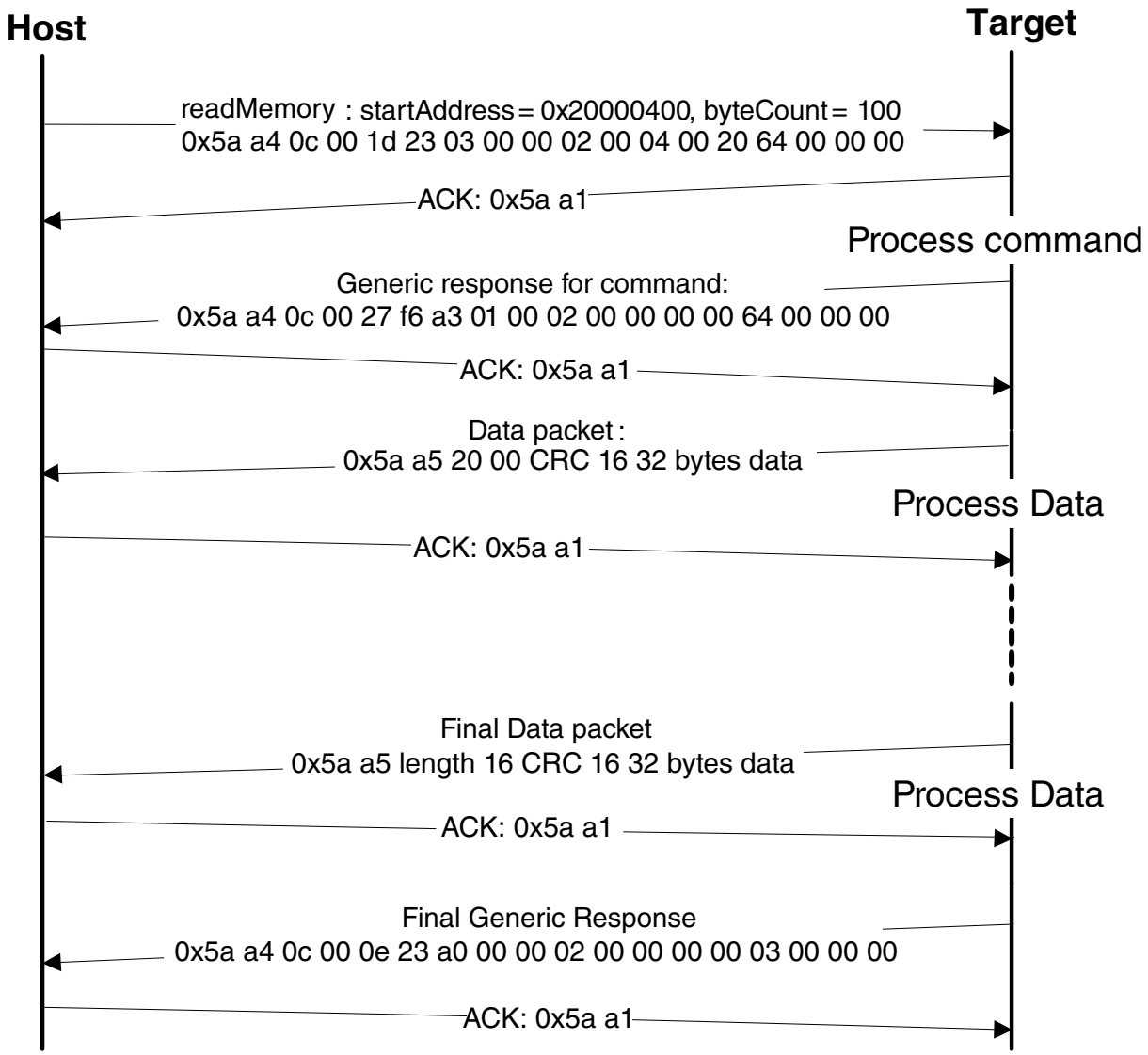


Figure 11-18. Command sequence for read memory

ReadMemory	Parameter	Value
Framing packet	Start byte	0x5A0xA4,
	packetType	kFramingPacketType_Command
	length	0x0C 0x00
	crc16	0x1D 0x23
Command packet	commandTag	0x03 - readMemory
	flags	0x00
	reserved	0x00
	parameterCount	0x02
	startAddress	0x20000400
	byteCount	0x00000064

Data Phase: The ReadMemory command has a data phase. Since the target (Kinetis Flashloader) works in slave mode, the host need pull data packets until the number of bytes of data specified in the byteCount parameter of ReadMemory command are received by host.

Response: The target (Kinetis Flashloader) will return a GenericResponse packet with a status code either set to kStatus_Success upon successful execution of the command, or set to an appropriate error status code.

11.3.6.13 Execute command

The execute command results in the flashloader setting the program counter to the code at the provided jump address, R0 to the provided argument, and a Stack pointer to the provided stack pointer address. Prior to the jump, the system is returned to the reset state.

The Jump address, function argument pointer, and stack pointer are the parameters required for the Execute command.

Table 11-43. Parameters for Execute Command

Byte #	Command
0 - 3	Jump address
4 - 7	Argument word
8 - 11	Stack pointer address

The Execute command has no data phase.

Response: Before executing the Execute command, the target (Kinetis Flashloader) will validate the parameters and return a GenericResponse packet with a status code either set to kStatus_Success or an appropriate error status code.

11.3.6.14 Reset command

The Reset command will result in flashloader resetting the chip.

The Reset command requires no parameters.

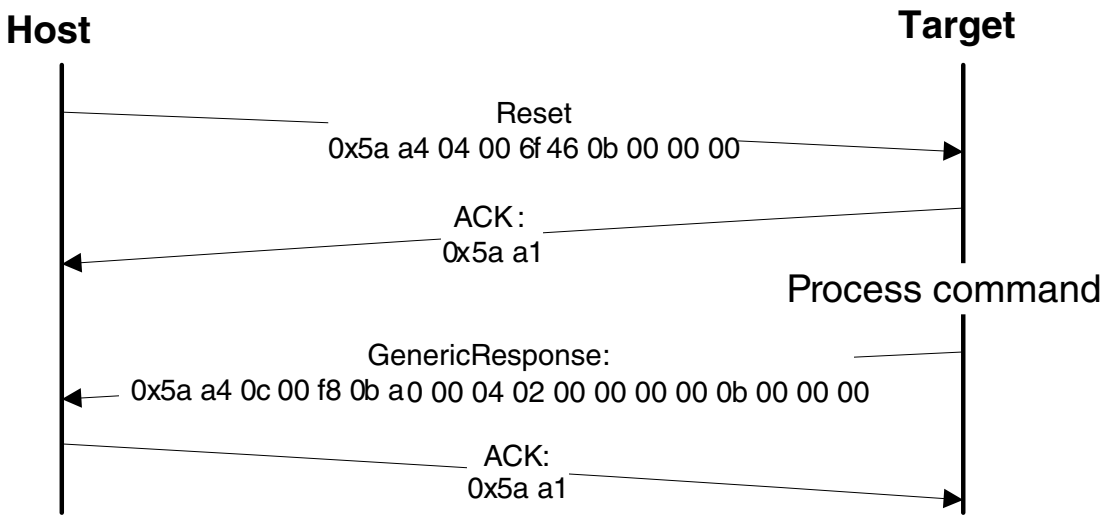


Figure 11-19. Protocol Sequence for Reset Command

Table 11-44. Reset Command Packet Format (Example)

Reset	Parameter	Value
Framing packet	start byte	0x5A
	packetType	0xA4, kFramingPacketType_Command
	length	0x04 0x00
	crc16	0x6F 0x46
Command packet	commandTag	0x0B - reset
	flags	0x00
	reserved	0x00
	parameterCount	0x00

The Reset command has no data phase.

Response: The target (Kinetis Flashloader) will return a GenericResponse packet with status code set to kStatus_Success, before resetting the chip.

11.4 Peripherals Supported

This section describes the peripherals supported by the Kinetis Flashloader.

11.4.1 I2C Peripheral

The Kinetis Flashloader supports loading data into flash via the I2C peripheral, where the I2C peripheral serves as the I2C slave. A 7-bit slave address is used during the transfer.

The Kinetis Flashloader uses 0x10 as the I2C slave address, and supports 400 kbps as the I2C baud rate.

Because the I2C peripheral serves as an I2C slave device, each transfer should be started by the host, and each outgoing packet should be fetched by the host.

- An incoming packet is sent by the host with a selected I2C slave address and the direction bit is set as write.
- An outgoing packet is read by the host with a selected I2C slave address and the direction bit is set as read.
- 0x00 will be sent as the response to host if the target is busy with processing or preparing data.

The following flow charts demonstrate the communication flow of how the host reads ping packet, ACK and response from the target.

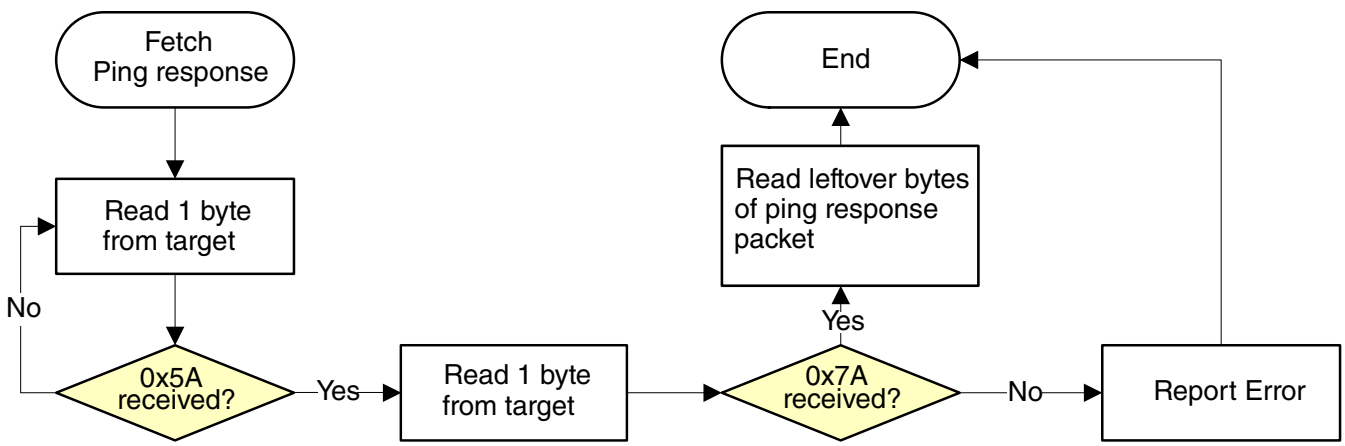


Figure 11-20. Host reads ping response from target via I2C

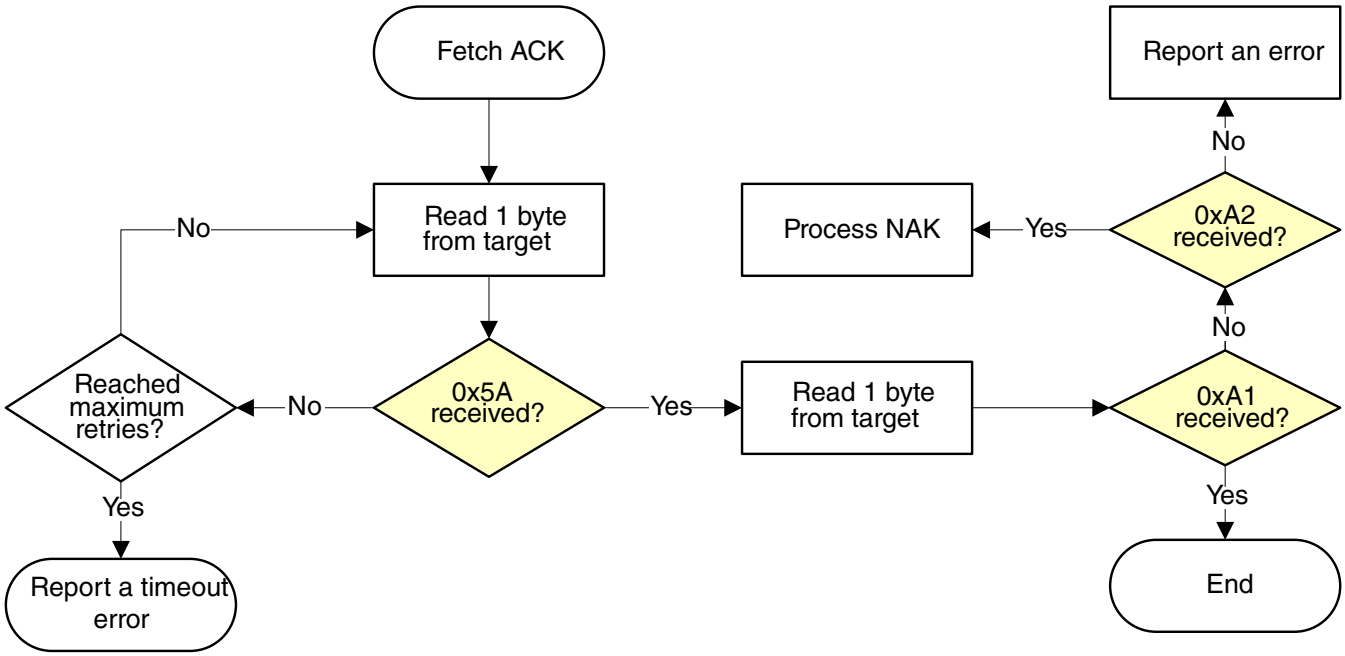


Figure 11-21. Host reads ACK packet from target via I2C

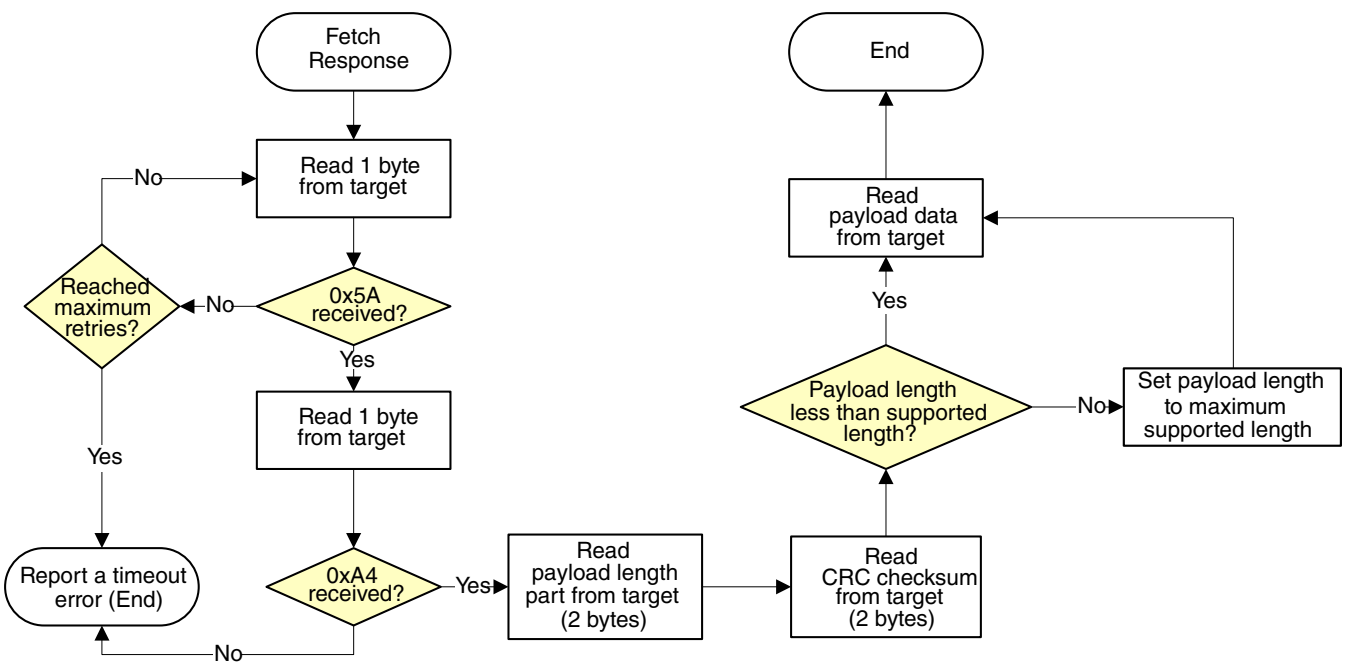


Figure 11-22. Host reads response from target via I2C

11.4.2 SPI Peripheral

The Kinetis Flashloader supports loading data into flash via the SPI peripheral, where the SPI peripheral serves as a SPI slave.

The Kinetis Flashloader supports 400 kbps as the SPI baud rate.

The SPI peripheral uses the following bus attributes:
- Clock Phase = 1 (Second Edge)
- Clock Polarity = 1 (Active Low)

Because the SPI peripheral serves as a SPI slave device, each transfer should be started by the host, and each outgoing packet should be fetched by the host.

The transfer on SPI is slightly different from I2C:
- Host will receive 1 byte after it sends out any byte.
- Received bytes should be ignored when host is sending out bytes to target
- Host starts reading bytes by sending 0x00s to target
- The byte 0x00 will be sent as response to host if target is under the following conditions:
  - Processing incoming packet
  - Preparing outgoing data
  - Received invalid data

The SPI bus configuration is:
- Phase = 1; data is sampled on rising edges
- Polarity = 1; idle is high
- MSB is transmitted first

For any transfer where the target does not have actual data to send, the target (slave) is responsible for ensuring that 0x00 bytes will be returned to the host (master). The host uses framing packets to identify real data and not "dummy" 0x00 bytes (which do not have framing packets).

The following flowcharts demonstrate how the host reads a ping response, an ACK and a command response from target via SPI.

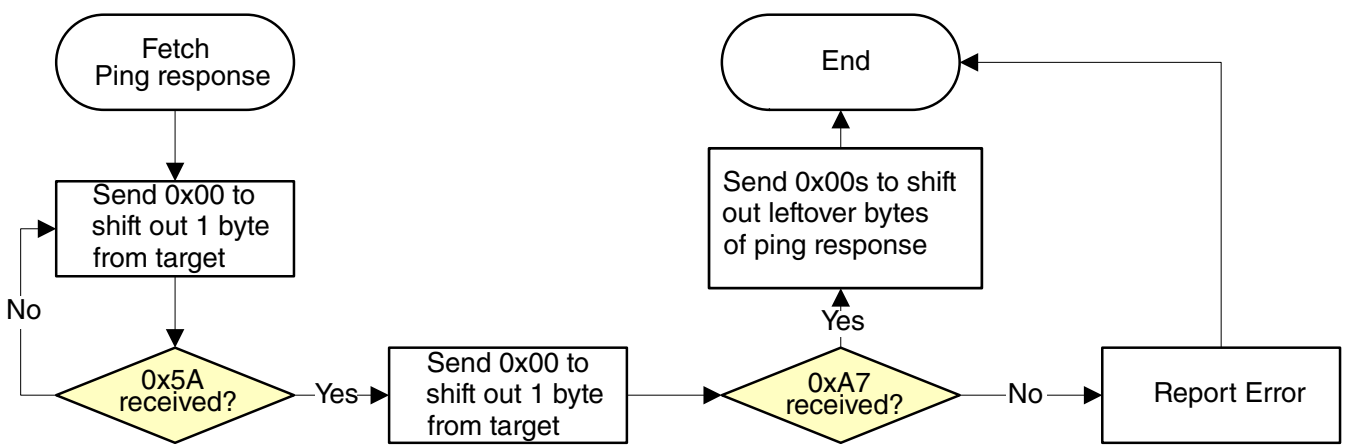


Figure 11-23. Host reads ping packet from target via SPI

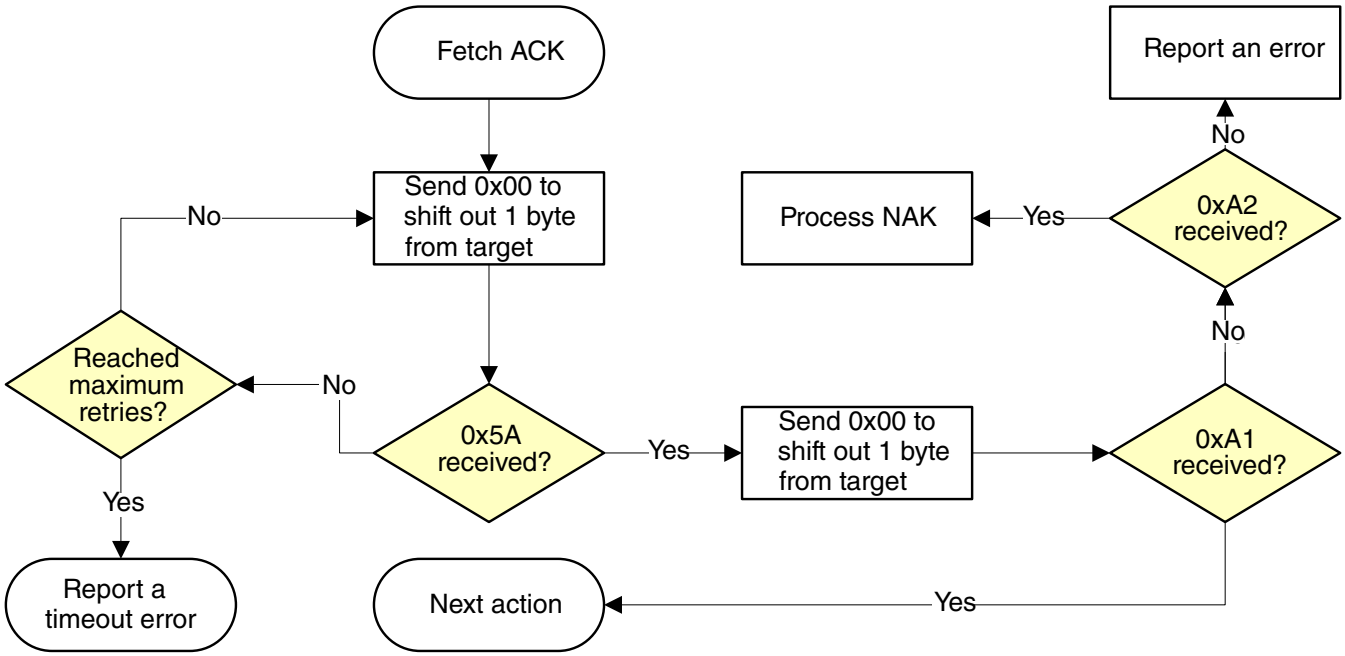


Figure 11-24. Host reads ACK from target via SPI

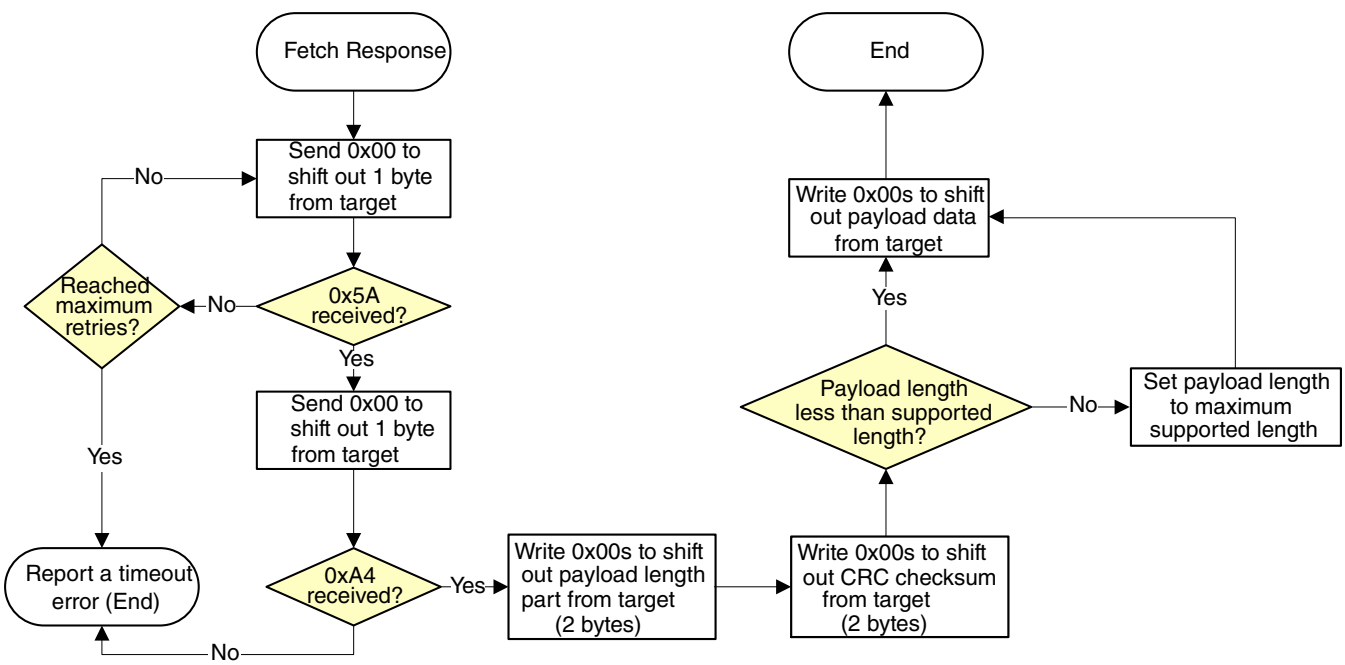


Figure 11-25. Host reads response from target via SPI

11.4.3 LPUART Peripheral

The Kinetis Flashloader integrates an autobaud detection algorithm for the LPUART peripheral, thereby providing flexible baud rate choices.

Autobaud feature: If LPUART*n* is used to connect to the flashloader, then the LPUART*n*_RX pin must be kept high and not left floating during the detection phase in order to comply with the autobaud detection algorithm. After the flashloader detects the ping packet (0x5A 0xA6) on LPUART*n*_RX, the flashloader firmware executes the autobaud sequence. If the baudrate is successfully detected, then the flashloader will send a ping packet response [(0x5A 0xA7), protocol version (4 bytes), protocol version options (2 bytes) and crc16 (2 bytes)] at the detected baudrate. The Kinetis Flashloader then enters a loop, waiting for flashloader commands via the LPUART peripheral.

NOTE

- The autobaud feature requires a ping packet with a higher accuracy (+/-3%), or the ping packet will be ignored as noise.
- The data bytes of the ping packet must be sent continuously (with no more than 80 ms between bytes) in a fixed LPUART transmission mode (8-bit data, no parity bit and 1 stop bit). If the bytes of the ping packet are sent one-by-one with more than 80 ms delay between them, then the autobaud detection algorithm may calculate an incorrect baud rate. In this case, the autobaud detection state machine should be reset.

Supported baud rates: The baud rate is closely related to the MCU core and system clock frequencies. Typical baud rates supported are 9600, 19200, 38400, 57600, and 115200.

Packet transfer: After autobaud detection succeeds, flashloader communications can take place over the LPUART peripheral. The following flow charts show:

- How the host detects an ACK from the target
- How the host detects a ping response from the target
- How the host detects a command response from the target

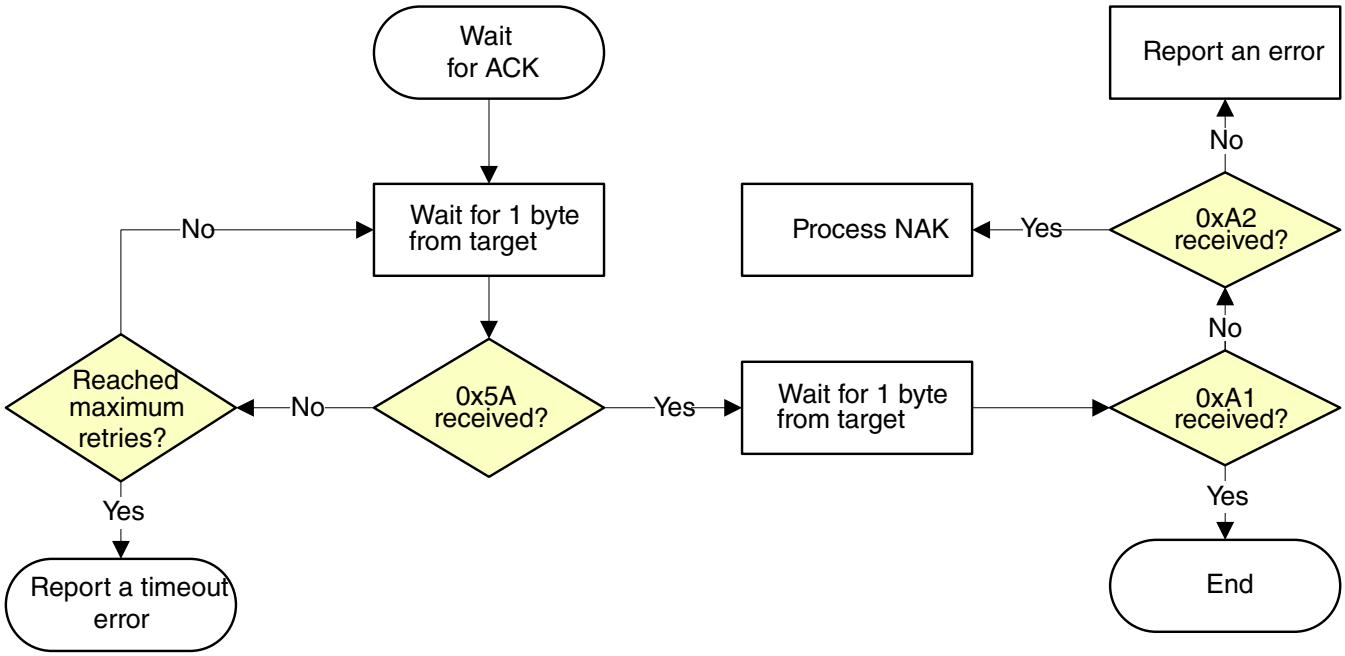


Figure 11-26. Host reads an ACK from target via LPUART

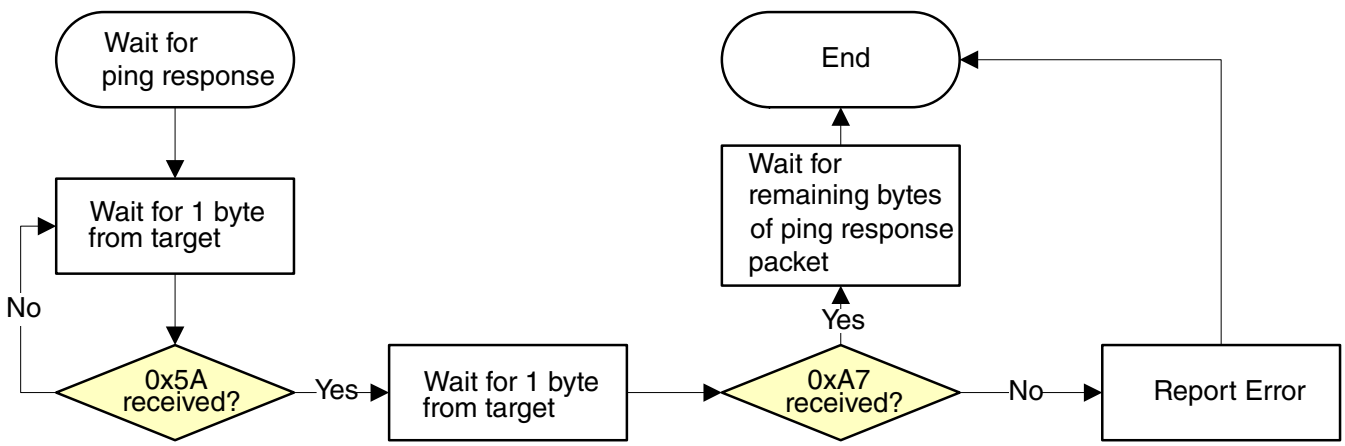


Figure 11-27. Host reads a ping response from target via LPUART

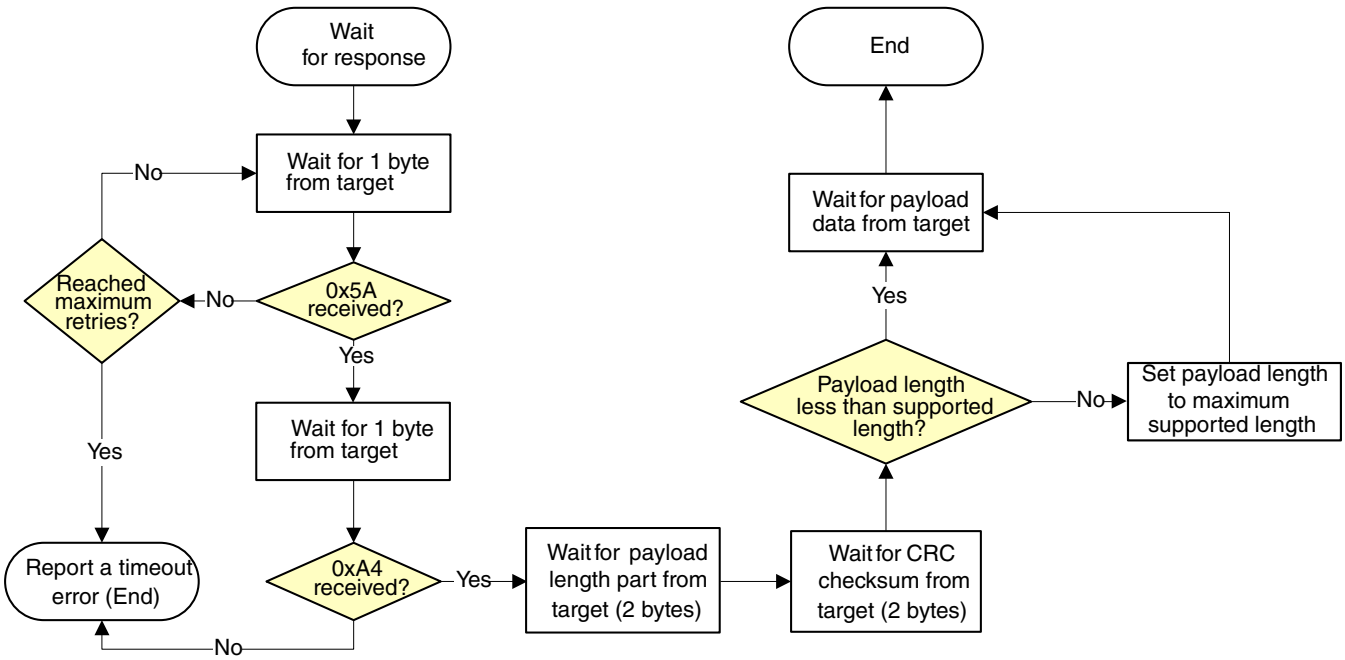


Figure 11-28. Host reads a command response from target via LPUART

11.4.4 CAN (or FlexCAN) Peripheral

The Kinetis Flashloader supports loading data into flash via the FlexCAN peripheral. Transfers to FlexCAN are supported at several predefined speeds:

- 125 kHz
- 250 kHz
- 500 kHz
- 1 MHz (the default transfer rate)

The host application must use one of the several supported speeds for FlexCAN. In Flashloader, it supports automatic speed detection within supported speeds. The Flashloader will enter the listen mode in the beginning with the initial speed (default speed 1 MHz). Once the host sends a ping to a specific node, it will generate traffic on the FlexCAN bus. Because the Flashloader is in a listen mode, it will be able to check if the local node speed is correct, by detecting errors.

- If there is an error, then some transfers may not be at the right speed.
- The Flashloader will change the speed setting and check again.
- If there is no error, it means that the transfer speed is correct, and it changes the settings back to normal receiving mode, to see if there is a package for this node.
- The host side should also have reasonable time tolerance during the automatic speed detection period. If there is a timeout, it means that there is no response from the specific node, or there is a real error (and it should report the error to the application).

The following flowcharts show how the host reads a ping packet, ACK and response from the target.

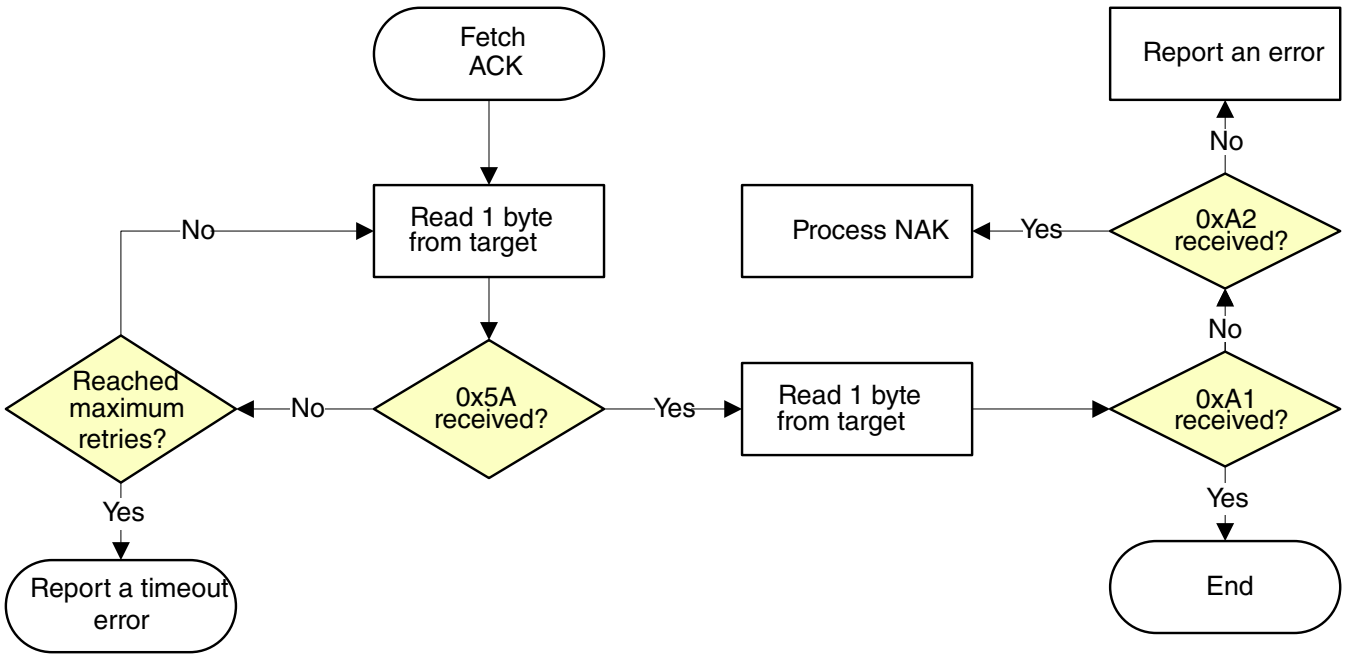


Figure 11-29. Host reads an ACK from target via FlexCAN

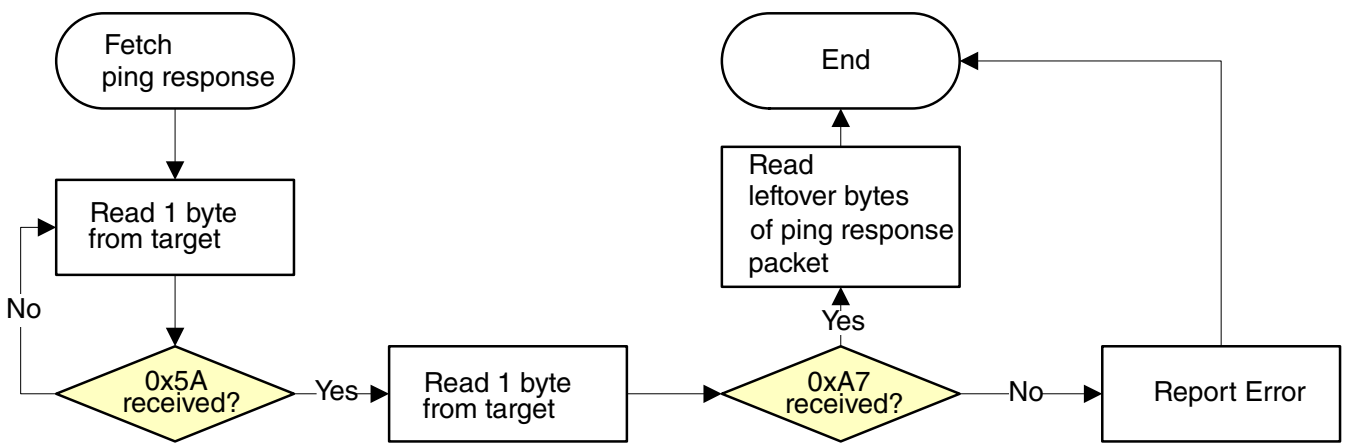


Figure 11-30. Host reads a ping response from target via FlexCAN

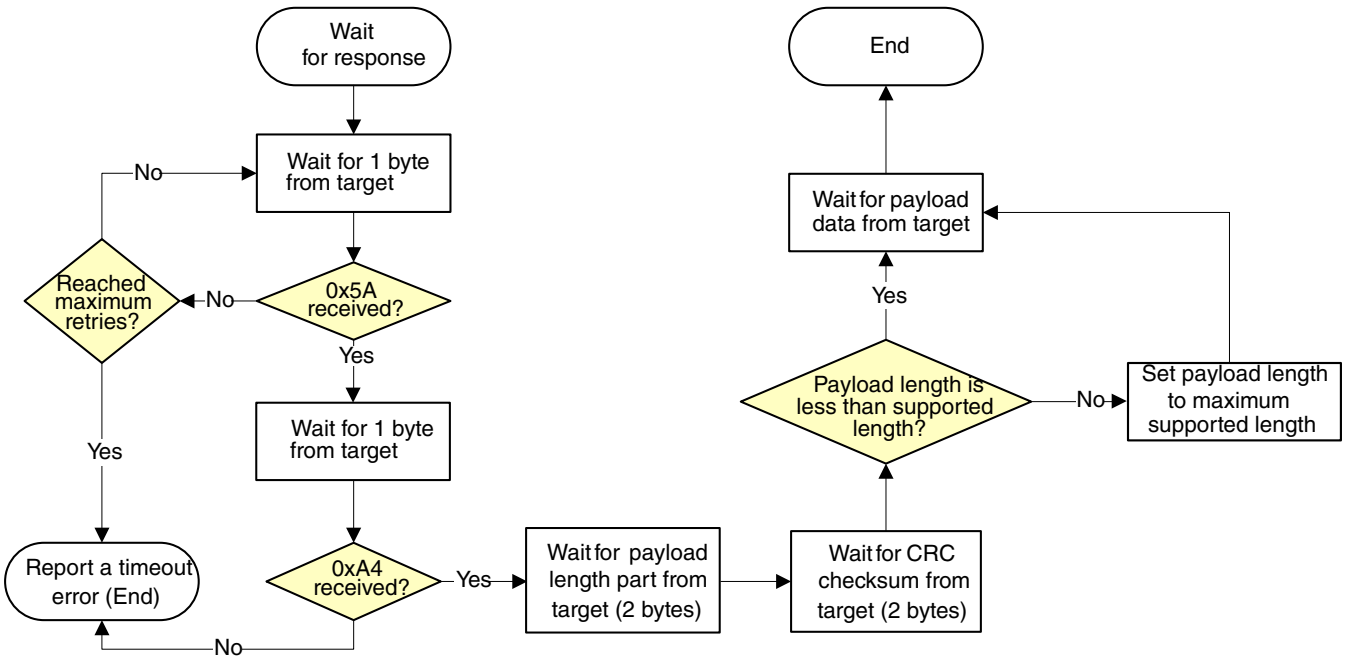


Figure 11-31. Host reads a command response from target via FlexCAN

11.5 Get/SetProperty Command Properties

This section lists the properties of the GetProperty and SetProperty commands.

Table 11-45. Properties used by Get/SetProperty Commands, sorted by Value

Property	Writable	Tag Value	Size	Description
CurrentVersion	No	01h	4	Current flashloader version.
AvailablePeripherals	No	02h	4	The set of peripherals supported on this chip.
FlashStartAddress	No	03h	4	Start address of program flash.
FlashSizeInBytes	No	04h	4	Size in bytes of program flash.
FlashSectorSize	No	05h	4	The size in bytes of one sector of program flash. This is the minimum erase size.
FlashBlockCount	No	06h	4	Number of blocks in the flash array.
AvailableCommands	No	07h	4	The set of commands supported by the flashloader.
VerifyWrites	Yes	0Ah	4	Controls whether the flashloader will verify writes to flash. VerifyWrites feature is enabled by default. 0 - No verification is done. 1 - Enable verification.
MaxPacketSize	No	0Bh	4	Maximum supported packet size for the currently active peripheral interface.

Table continues on the next page...

Table 11-45. Properties used by Get/SetProperty Commands, sorted by Value (continued)

Property	Writable	Tag Value	Size	Description
ReservedRegions	No	0Ch	16	List of memory regions reserved by the flashloader. Returned as value pairs (<start-address-of-region>, <end-address-of-region>). • If HasDataPhase flag is not set, then the Response packet parameter count indicates the number of pairs. • If HasDataPhase flag is set, then the second parameter is the number of bytes in the data phase.
RAMStartAddress	No	0Eh	4	Start address of RAM segment. The first parameter to GetProperty command identifies the segment. See the device specific memory map for number of RAM segments the device contains.
RAMSizeInBytes	No	0Fh	4	Size in bytes of RAM segment. The first parameter to GetProperty command identifies the segment. See the device specific memory map for number of RAM segments the device contains.
SystemDeviceId	No	10h	4	Value of the Kinetis System Device Identification register.
FlashSecurityState	No	11h	4	Indicates whether Flash security is enabled 0 - Flash security is disabled 1 - Flash security is enabled
UniqueDeviceId	No	12h	16	Unique device identification, value of Kinetis Unique Identification registers (16 for K series devices, 12 for KL series devices)
FacSupport	No	13h	4	FAC (Flash Access Control) support flag 0 - FAC not supported 1 - FAC supported
FlashAcessSegmentSize	No	14h	4	The size in bytes of 1 segment of flash
FlashAcessSegmentCount	No	15h	4	FAC segment count (The count of flash access segments within the flash model.)
FlashReadMargin	Yes	16h	4	The margin level setting for flash erase and program verify commands. 0 = Normal 1 = User (default) 2 = Factory
TargetVersion	No	18h	4	SoC target build version number

11.5.1 Property Definitions

Get/Set property definitions are provided in this section.

11.5.1.1 CurrentVersion Property

The value of this property is a 4-byte structure containing the current version of the flashloader.

Table 11-46. Fields of CurrentVersion property:

Bits	[31:24]	[23:16]	[15:8]	[7:0]
Field	Name = 'K' (0x4B)	Major version	Minor version	Bugfix version

11.5.1.2 AvailablePeripherals Property

The value of this property is a bitfield that lists the peripherals supported by the flashloader and the hardware on which it is running.

Table 11-47. Peripheral bits:

Bit	[31:7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Peripheral	Reserved	Reserved	Reserved	Reserved	CAN Slave	SPI Slave	I2C Slave	LPUART

If the peripheral is available, then the corresponding bit will be set in the property value. All reserved bits must be set to 0.

11.5.1.3 AvailableCommands Property

This property value is a bitfield with set bits indicating the commands enabled in the flashloader. Only commands that can be sent from the host to the target are listed in the bitfield. Response commands such as GenericResponse are excluded.

The bit number that identifies whether a command is present is the command's tag value minus 1. 1 is subtracted from the command tag because the lowest command tag value is 0x01. To get the bit mask for a given command, use this expression:

```
mask = 1 << (tag - 1)
```

Table 11-48. Command bits:

Bit	[31:18]	[17]	[16]	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Command	Reserved	Reserved	Reserved	FlashReadResource	FlashReadOnce	FlashProgramOnce	FlashEraseAllUnsecure	SetProperty	Reset	Call	Execute	Reserved	GetProperty	Reserved	FillMemory	WriteMemory	ReadMemory	FlashEraseRegion	FlashEraseAll

11.5.1.4 TargetVersion Property

The value of this property is a 4-byte structure containing the target version of the flashloader.

Table 11-49. Fields of TargetVersion property:

Bits	[31:24]	[23:16]	[15:8]	[7:0]
Field	Name = 'T' (0x4B)	Major version	Minor version	Bugfix version

11.6 Kinetis Flashloader Status Error Codes

This section describes the status error codes that the Kinetis Flashloader returns to the host.

Table 11-50. Kinetis Flashloader Status Error Codes, sorted by Value

Error Code	Value	Description
kStatus_Success	0	Operation succeeded without error.
kStatus_Fail	1	Operation failed with a generic error.
kStatus_ReadOnly	2	Requested value cannot be changed because it is read-only.
kStatus_OutOfRange	3	Requested value is out of range.
kStatus_InvalidArgument	4	The requested command's argument is undefined.
kStatus_Timeout	5	A timeout occurred.
kStatus_FlashSizeError	100	Not used.
kStatus_FlashAlignmentError	101	Address or length does not meet required alignment.

Table continues on the next page...

Table 11-50. Kinetis Flashloader Status Error Codes, sorted by Value (continued)

Error Code	Value	Description
kStatus_FlashAddressError	102	Address or length is outside addressable memory.
kStatus_FlashAccessError	103	The FTFA_FSTAT[ACCERR] bit is set.
kStatus_FlashProtectionViolation	104	The FTFA_FSTAT[FPVIOL] bit is set.
kStatus_FlashCommandFailure	105	The FTFA_FSTAT[MGSTAT0] bit is set.
kStatus_FlashUnknownProperty	106	Unknown Flash property.
kStatus_FlashEraseKeyError	107	The key provided does not match the programmed flash key.
kStatus_FlashRegionExecuteOnly	108	The area of flash is protected as execute only.
kStatus_I2C_SlaveTxUnderrun	200	I2C Slave TX Underrun error.
kStatus_I2C_SlaveRxOverrun	201	I2C Slave RX Overrun error.
kStatus_I2C_AribtrationLost	202	I2C Arbitration Lost error.
kStatus_SPI_SlaveTxUnderrun	300	SPI Slave TX Underrun error.
kStatus_SPI_SlaveRxOverrun	301	SPI Slave RX Overrun error.
kStatus_SPI_Timeout	302	SPI tranfser timed out.
kStatus_SPI_Busy	303	SPI instance is already busy performing a transfer.
kStatus_SPI_NoTransferInProgress	304	Attempt to abort a transfer when no transfer was in progress.
kStatus_UnknownCommand	10000	The requested command value is undefined.
kStatus_SecurityViolation	10001	Command is disallowed because flash security is enabled.
kStatus_AbortDataPhase	10002	Abort the data phase early.
kStatusMemoryRangeInvalid	10200	Memory range conflicts with a protected region.
kStatus_UnknownProperty	10300	The requested property value is undefined.
kStatus_ReadOnlyProperty	10301	The requested property value cannot be written.
kStatus_InvalidPropertyValue	10302	The specified property value is invalid.
kStatus_AppCrcCheckPassed	10400	CRC check is valid and passed.
kStatus_AppCrcCheckFailed	10401	CRC check is valid but failed.
kStatus_AppCrcCheckInactive	10402	CRC check is inactive.
kStatus_AppCrcCheckInvalid	10403	CRC check is invalid, because the BCA is invalid or the CRC parameters are unset (all 0xFF bytes).
kStatus_AppCrcCheckOutOfRange	10404	CRC check is valid but addresses are out of range.

Chapter 12
Port control and interrupt (PORT)

12.1 Introduction

12.1.1 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.

12.1.2 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
  - Peripheral trigger output (active high, low) configured per pin
- Port control
  - Individual pull control fields with pullup, pulldown, and pull-disable support
  - Individual slew rate field supporting fast and slow slew rates
  - Individual mux control field supporting analog or pin disabled, GPIO, and up to 14 chip-specific digital functions
  - Pad configuration fields are functional in all digital pin muxing modes

12.1.3 Modes of operation

12.1.3.1 Run mode

In Run mode, the PORT operates normally.

12.1.3.2 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.

12.1.3.3 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.

In Stop mode, the PORT configuration is static.

12.1.3.4 Debug mode

In Debug mode, PORT operates normally.

12.2 External signal description

The table found here describes the PORT external signal.

Table 12-1. 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.

12.2.1 Detailed signal description

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

Table 12-2. PORT interface—detailed signal description

Signal	I/O	Description	
PORTx[31:0]	I/O	External interrupt.	
		State meaning	Asserted—pin is logic 1. Negated—pin is logic 0.
		Timing	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.

12.3 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.

12.3.1 PORT register descriptions

12.3.1.1 PORT Memory map

PORTA base address: 4004_9000h

PORTB base address: 4004_A000h

PORTC base address: 4004_B000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Pin Control Register 0 (PCR0)	32	RW	See description.
4h	Pin Control Register 1 (PCR1)	32	RW	See description.
8h	Pin Control Register 2 (PCR2)	32	RW	0000_0005h
Ch	Pin Control Register 3 (PCR3)	32	RW	See description.

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
10h - 14h	Pin Control Register a (PCR4 - PCR5)	32	RW	0000_0005h
18h	Pin Control Register 6 (PCR6)	32	RW	0000_0005h
1Ch	Pin Control Register 7 (PCR7)	32	RW	0000_0001h
40h	Pin Control Register 16 (PCR16)	32	RW	See description.
44h	Pin Control Register 17 (PCR17)	32	RW	See description.
48h	Pin Control Register 18 (PCR18)	32	RW	See description.
4Ch	Pin Control Register 19 (PCR19)	32	RW	0000_0001h
80h	Global Pin Control Low Register (GPCLR)	32	WORZ	0000_0000h
84h	Global Pin Control High Register (GPCHR)	32	WORZ	0000_0000h
88h	Global Interrupt Control Low Register (GICLR)	32	WORZ	0000_0000h
8Ch	Global Interrupt Control High Register (GICHR)	32	WORZ	0000_0000h
A0h	Interrupt Status Flag Register (ISFR)	32	W1C	0000_0000h

12.3.1.2 Pin Control Register 0 (PCR0)

12.3.1.2.1 Offset

Register	Offset
PCR0	0h

12.3.1.2.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

12.3.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
W								W1C								
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0		MUX				0	DSE	0	0	0	SRE	PE	PS
W																
Reset	See Register reset values.															

12.3.1.2.4 Register reset values

Register	Reset value
PCR0	PORTA: 0000_0F03h PORTB: 0000_0001h PORTC: 0000_0005h

12.3.1.2.5 Fields

Field	Function
31 —	Reserved.
30-25 —	Reserved
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected. 1b - 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 —	Reserved.
22-20 —	Reserved
19-16 IRQC	Interrupt Configuration

Table continues on the next page...

Field	Function
	The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge. 1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.
15 —	Reserved.
14 —	Reserved.
13-12 —	Reserved
11-8 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: 0000b - Pin disabled (Alternative 0) (analog). 0001b - Alternative 1 (GPIO). 0010b - Alternative 2 (chip-specific). 0011b - Alternative 3 (chip-specific). 0100b - Alternative 4 (chip-specific). 0101b - Alternative 5 (chip-specific). 0110b - Alternative 6 (chip-specific). 0111b - Alternative 7 (chip-specific). 1000b - Alternative 8 (chip-specific). 1001b - Alternative 9 (chip-specific). 1010b - Alternative 10 (chip-specific). 1011b - Alternative 11 (chip-specific). 1100b - Alternative 12 (chip-specific). 1101b - Alternative 13 (chip-specific). 1110b - Alternative 14 (chip-specific). 1111b - Alternative 15 (chip-specific).
7 —	Reserved
6 DSE	Drive Strength Enable Drive strength configuration is valid in all digital pin muxing modes.

Table continues on the next page...

Field	Function
	NOTE: This field is not supported in every instance. The following table includes only supported registers. **Field supported in** / **Field not supported in** — / PORTA_PCR0 PORTB_PCR0 / — PORTC_PCR0 / — 0b - Low drive strength is configured on the corresponding pin, if pin is configured as a digital output. 1b - High drive strength is configured on the corresponding pin, if pin is configured as a digital output.
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 SRE	Slew Rate Enable Slew rate configuration is valid in all digital pin muxing modes. 0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output. 1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0b - Internal pull resistor is not enabled on the corresponding pin. 1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

12.3.1.3 Pin Control Register 1 (PCR1)

12.3.1.3.1 Offset

Register	Offset
PCR1	4h

12.3.1.3.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

12.3.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
W								W1C								
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0		MUX				0	DSE	0	0	0	SRE	PE	PS
W																
Reset	See Register reset values.															

12.3.1.3.4 Register reset values

Register	Reset value
PCR1	PORTA: 0000_0F06h PORTB,PORTC: 0000_0005h

12.3.1.3.5 Fields

Field	Function
31 —	Reserved.
30-25	Reserved

Table continues on the next page...

Field	Function
—	
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected. 1b - 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 —	Reserved.
22-20 —	Reserved
19-16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge. 1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.
15 —	Reserved.
14 —	Reserved.
13-12 —	Reserved
11-8 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: 0000b - Pin disabled (Alternative 0) (analog). 0001b - Alternative 1 (GPIO). 0010b - Alternative 2 (chip-specific). 0011b - Alternative 3 (chip-specific).

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>0100b - Alternative 4 (chip-specific).
0101b - Alternative 5 (chip-specific).
0110b - Alternative 6 (chip-specific).
0111b - Alternative 7 (chip-specific).
1000b - Alternative 8 (chip-specific).
1001b - Alternative 9 (chip-specific).
1010b - Alternative 10 (chip-specific).
1011b - Alternative 11 (chip-specific).
1100b - Alternative 12 (chip-specific).
1101b - Alternative 13 (chip-specific).
1110b - Alternative 14 (chip-specific).
1111b - Alternative 15 (chip-specific).</td></tr>
<tr><td>7
—</td><td>Reserved</td></tr>
<tr><td>6
DSE</td><td>Drive Strength Enable
Drive strength configuration is valid in all digital pin muxing modes.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>—</td><td>PORTA_PCR1</td></tr><tr><td>PORTB_PCR1</td><td>—</td></tr><tr><td>PORTC_PCR1</td><td>—</td></tr></table>
0b - Low drive strength is configured on the corresponding pin, if pin is configured as a digital output.
1b - High drive strength is configured on the corresponding pin, if pin is configured as a digital output.</td></tr>
<tr><td>5
—</td><td>Reserved</td></tr>
<tr><td>4
—</td><td>Reserved</td></tr>
<tr><td>3
—</td><td>Reserved</td></tr>
<tr><td>2
SRE</td><td>Slew Rate Enable
Slew rate configuration is valid in all digital pin muxing modes.
0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.</td></tr>
<tr><td>1
PE</td><td>Pull Enable
Pull configuration is valid in all digital pin muxing modes.
0b - Internal pull resistor is not enabled on the corresponding pin.
1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.</td></tr>
<tr><td>0</td><td>Pull Select</td></tr>
</table>

Field	Function
PS	Pull configuration is valid in all digital pin muxing modes. 0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

12.3.1.4 Pin Control Register 2 (PCR2)

12.3.1.4.1 Offset

Register	Offset
PCR2	8h

12.3.1.4.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

12.3.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
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	0	0	0		MUX				0	DSE	0	0	0	SRE	PE	PS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

12.3.1.4.4 Fields

Field	Function
31 —	Reserved.
30-25 —	Reserved
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected. 1b - 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 —	Reserved.
22-20 —	Reserved
19-16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge. 1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.
15 —	Reserved.
14 —	Reserved.
13-12 —	Reserved
11-8	Pin Mux Control

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>MUX</td><td>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:

0000b - Pin disabled (Alternative 0) (analog).
0001b - Alternative 1 (GPIO).
0010b - Alternative 2 (chip-specific).
0011b - Alternative 3 (chip-specific).
0100b - Alternative 4 (chip-specific).
0101b - Alternative 5 (chip-specific).
0110b - Alternative 6 (chip-specific).
0111b - Alternative 7 (chip-specific).
1000b - Alternative 8 (chip-specific).
1001b - Alternative 9 (chip-specific).
1010b - Alternative 10 (chip-specific).
1011b - Alternative 11 (chip-specific).
1100b - Alternative 12 (chip-specific).
1101b - Alternative 13 (chip-specific).
1110b - Alternative 14 (chip-specific).
1111b - Alternative 15 (chip-specific).</td></tr>
<tr><td>7

—</td><td>Reserved</td></tr>
<tr><td>6

DSE</td><td>Drive Strength Enable

Drive strength configuration is valid in all digital pin muxing modes.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>—</td><td>PORTA_PCR2</td></tr><tr><td>—</td><td>PORTB_PCR2</td></tr><tr><td>PORTC_PCR2</td><td>—</td></tr></table>
0b - Low drive strength is configured on the corresponding pin, if pin is configured as a digital output.
1b - High drive strength is configured on the corresponding pin, if pin is configured as a digital output.</td></tr>
<tr><td>5

—</td><td>Reserved</td></tr>
<tr><td>4

—</td><td>Reserved</td></tr>
<tr><td>3

—</td><td>Reserved</td></tr>
<tr><td>2

SRE</td><td>Slew Rate Enable

Slew rate configuration is valid in all digital pin muxing modes.

0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.</td></tr>
</table>

Table continues on the next page...

Field	Function
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0b - Internal pull resistor is not enabled on the corresponding pin. 1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

12.3.1.5 Pin Control Register 3 (PCR3)

12.3.1.5.1 Offset

Register	Offset
PCR3	Ch

12.3.1.5.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
—	PORTA _PCR3
PORTB _PCR3	—
PORTC _PCR3	—

12.3.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
W								W1C								
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0		MUX				0	DSE	0	0	0	SRE	PE	PS
W																
Reset	See Register reset values.															

12.3.1.5.4 Register reset values

Register	Reset value
PCR3	PORTA: Register not supported PORTB: 0000_0001h PORTC: 0000_0005h

12.3.1.5.5 Fields

Field	Function
31 —	Reserved.
30-25 —	Reserved
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected.

Table continues on the next page...

Field	Function
	1b - 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 ——	Reserved.
22-20 ——	Reserved
19-16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge. 1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.
15 ——	Reserved.
14 ——	Reserved.
13-12 ——	Reserved
11-8 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: 0000b - Pin disabled (Alternative 0) (analog). 0001b - Alternative 1 (GPIO). 0010b - Alternative 2 (chip-specific). 0011b - Alternative 3 (chip-specific). 0100b - Alternative 4 (chip-specific). 0101b - Alternative 5 (chip-specific). 0110b - Alternative 6 (chip-specific). 0111b - Alternative 7 (chip-specific). 1000b - Alternative 8 (chip-specific).

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>1001b - Alternative 9 (chip-specific).
1010b - Alternative 10 (chip-specific).
1011b - Alternative 11 (chip-specific).
1100b - Alternative 12 (chip-specific).
1101b - Alternative 13 (chip-specific).
1110b - Alternative 14 (chip-specific).
1111b - Alternative 15 (chip-specific).</td></tr>
<tr><td>7
—</td><td>Reserved</td></tr>
<tr><td>6
DSE</td><td>Drive Strength Enable
Drive strength configuration is valid in all digital pin muxing modes.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>—</td><td>PORTB_PCR3</td></tr><tr><td>PORTC_PCR3</td><td>—</td></tr></table>
0b - Low drive strength is configured on the corresponding pin, if pin is configured as a digital output.
1b - High drive strength is configured on the corresponding pin, if pin is configured as a digital output.</td></tr>
<tr><td>5
—</td><td>Reserved</td></tr>
<tr><td>4
—</td><td>Reserved</td></tr>
<tr><td>3
—</td><td>Reserved</td></tr>
<tr><td>2
SRE</td><td>Slew Rate Enable
Slew rate configuration is valid in all digital pin muxing modes.
0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.</td></tr>
<tr><td>1
PE</td><td>Pull Enable
Pull configuration is valid in all digital pin muxing modes.
0b - Internal pull resistor is not enabled on the corresponding pin.
1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.</td></tr>
<tr><td>0
PS</td><td>Pull Select
Pull configuration is valid in all digital pin muxing modes.
0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set.
1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.</td></tr>
</table>

12.3.1.6 Pin Control Register a (PCR4 - PCR5)

12.3.1.6.1 Offset

Register	Offset
PCR4	10h
PCR5	14h

12.3.1.6.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
—	PORTA _PCR4– PCR5
—	PORTB _PCR4– PCR5
PORTC _PCR4– PCR5	—

12.3.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
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	0	0	0		MUX				0	0	0	0	0	SRE	PE	PS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

12.3.1.6.4 Fields

Field	Function
31 —	Reserved.
30-25 —	Reserved
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected. 1b - 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 —	Reserved.
22-20 —	Reserved
19-16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge.

Table continues on the next page...

Field	Function
	1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.
15 — 	Reserved.
14 — 	Reserved.
13-12 — 	Reserved
11-8 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: 0000b - Pin disabled (Alternative 0) (analog). 0001b - Alternative 1 (GPIO). 0010b - Alternative 2 (chip-specific). 0011b - Alternative 3 (chip-specific). 0100b - Alternative 4 (chip-specific). 0101b - Alternative 5 (chip-specific). 0110b - Alternative 6 (chip-specific). 0111b - Alternative 7 (chip-specific). 1000b - Alternative 8 (chip-specific). 1001b - Alternative 9 (chip-specific). 1010b - Alternative 10 (chip-specific). 1011b - Alternative 11 (chip-specific). 1100b - Alternative 12 (chip-specific). 1101b - Alternative 13 (chip-specific). 1110b - Alternative 14 (chip-specific). 1111b - Alternative 15 (chip-specific).
7 — 	Reserved
6 — 	Reserved
5 — 	Reserved
4 — 	Reserved
3 — 	Reserved
2 SRE 	Slew Rate Enable Slew rate configuration is valid in all digital pin muxing modes. 0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output.

Table continues on the next page...

Field	Function
	1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0b - Internal pull resistor is not enabled on the corresponding pin. 1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

12.3.1.7 Pin Control Register 6 (PCR6)

12.3.1.7.1 Offset

Register	Offset
PCR6	18h

12.3.1.7.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
—	PORTA _PCR6
—	PORTB _PCR6
PORTC _PCR6	—

12.3.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
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	0	0	0		MUX				0	DSE	0	0	0	SRE	PE	PS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

12.3.1.7.4 Fields

Field	Function
31 —	Reserved.
30-25 —	Reserved
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected. 1b - 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 —	Reserved.
22-20 —	Reserved
19-16 IRQC	Interrupt Configuration

Table continues on the next page...

Field	Function
	The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge. 1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.
15 —	Reserved.
14 —	Reserved.
13-12 —	Reserved
11-8 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: 0000b - Pin disabled (Alternative 0) (analog). 0001b - Alternative 1 (GPIO). 0010b - Alternative 2 (chip-specific). 0011b - Alternative 3 (chip-specific). 0100b - Alternative 4 (chip-specific). 0101b - Alternative 5 (chip-specific). 0110b - Alternative 6 (chip-specific). 0111b - Alternative 7 (chip-specific). 1000b - Alternative 8 (chip-specific). 1001b - Alternative 9 (chip-specific). 1010b - Alternative 10 (chip-specific). 1011b - Alternative 11 (chip-specific). 1100b - Alternative 12 (chip-specific). 1101b - Alternative 13 (chip-specific). 1110b - Alternative 14 (chip-specific). 1111b - Alternative 15 (chip-specific).
7 —	Reserved
6 DSE	Drive Strength Enable Drive strength configuration is valid in all digital pin muxing modes.

Table continues on the next page...

Field	Function
	0b - Low drive strength is configured on the corresponding pin, if pin is configured as a digital output. 1b - High drive strength is configured on the corresponding pin, if pin is configured as a digital output.
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 SRE	Slew Rate Enable Slew rate configuration is valid in all digital pin muxing modes. 0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output. 1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0b - Internal pull resistor is not enabled on the corresponding pin. 1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

12.3.1.8 Pin Control Register 7 (PCR7)

12.3.1.8.1 Offset

Register	Offset
PCR7	1Ch

12.3.1.8.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
—	PORTA_PCR7
—	PORTB_PCR7
PORTC_PCR7	—

12.3.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
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	0	0	0		MUX				0	DSE	0	0	0	SRE	PE	PS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

12.3.1.8.4 Fields

Field	Function
31 —	Reserved.
30-25 —	Reserved
24	Interrupt Status Flag

Table continues on the next page...

Field	Function
ISF	The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected. 1b - 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 —	Reserved.
22-20 —	Reserved
19-16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge. 1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.
15 —	Reserved.
14 —	Reserved.
13-12 —	Reserved
11-8 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: 0000b - Pin disabled (Alternative 0) (analog). 0001b - Alternative 1 (GPIO). 0010b - Alternative 2 (chip-specific). 0011b - Alternative 3 (chip-specific). 0100b - Alternative 4 (chip-specific). 0101b - Alternative 5 (chip-specific). 0110b - Alternative 6 (chip-specific).

Table continues on the next page...

Field	Function
	0111b - Alternative 7 (chip-specific). 1000b - Alternative 8 (chip-specific). 1001b - Alternative 9 (chip-specific). 1010b - Alternative 10 (chip-specific). 1011b - Alternative 11 (chip-specific). 1100b - Alternative 12 (chip-specific). 1101b - Alternative 13 (chip-specific). 1110b - Alternative 14 (chip-specific). 1111b - Alternative 15 (chip-specific).
7 ——	Reserved
6 DSE	Drive Strength Enable Drive strength configuration is valid in all digital pin muxing modes. 0b - Low drive strength is configured on the corresponding pin, if pin is configured as a digital output. 1b - High drive strength is configured on the corresponding pin, if pin is configured as a digital output.
5 ——	Reserved
4 ——	Reserved
3 ——	Reserved
2 SRE	Slew Rate Enable Slew rate configuration is valid in all digital pin muxing modes. 0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output. 1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0b - Internal pull resistor is not enabled on the corresponding pin. 1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

12.3.1.9 Pin Control Register 16 (PCR16)

12.3.1.9.1 Offset

Register	Offset
PCR16	40h

12.3.1.9.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

12.3.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
W								W1C								
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0			MUX			0	0	0	0	0	SRE	PE	PS
W																
Reset	See Register reset values.															

12.3.1.9.4 Register reset values

Register	Reset value
PCR16	PORTA: 0000_0001h PORTB: 0000_0005h PORTC: 0000_0001h

12.3.1.9.5 Fields

Field	Function
31 —	Reserved.
30-25 —	Reserved
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected. 1b - 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 —	Reserved.
22-20 —	Reserved
19-16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge. 1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.
15 —	Reserved.
14 —	Reserved.
13-12 —	Reserved
11-8 MUX	Pin Mux Control

Table continues on the next page...

Field	Function
	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: 0000b - Pin disabled (Alternative 0) (analog). 0001b - Alternative 1 (GPIO). 0010b - Alternative 2 (chip-specific). 0011b - Alternative 3 (chip-specific). 0100b - Alternative 4 (chip-specific). 0101b - Alternative 5 (chip-specific). 0110b - Alternative 6 (chip-specific). 0111b - Alternative 7 (chip-specific). 1000b - Alternative 8 (chip-specific). 1001b - Alternative 9 (chip-specific). 1010b - Alternative 10 (chip-specific). 1011b - Alternative 11 (chip-specific). 1100b - Alternative 12 (chip-specific). 1101b - Alternative 13 (chip-specific). 1110b - Alternative 14 (chip-specific). 1111b - Alternative 15 (chip-specific).
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 SRE	Slew Rate Enable Slew rate configuration is valid in all digital pin muxing modes. 0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output. 1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0b - Internal pull resistor is not enabled on the corresponding pin. 1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

12.3.1.10 Pin Control Register 17 (PCR17)

12.3.1.10.1 Offset

Register	Offset
PCR17	44h

12.3.1.10.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

12.3.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
W								W1C								
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0			MUX			0	DSE	0	0	0	SRE	PE	PS
W																
Reset	See Register reset values.															

12.3.1.10.4 Register reset values

Register	Reset value
PCR17	PORTA: 0000_0001h PORTB: 0000_0005h PORTC: 0000_0001h

12.3.1.10.5 Fields

Field	Function
31 —	Reserved.
30-25 —	Reserved
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected. 1b - 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 —	Reserved.
22-20 —	Reserved
19-16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge. 1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>15
—</td><td>Reserved.</td></tr>
<tr><td>14
—</td><td>Reserved.</td></tr>
<tr><td>13-12
—</td><td>Reserved</td></tr>
<tr><td>11-8
MUX</td><td>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:
0000b - Pin disabled (Alternative 0) (analog).
0001b - Alternative 1 (GPIO).
0010b - Alternative 2 (chip-specific).
0011b - Alternative 3 (chip-specific).
0100b - Alternative 4 (chip-specific).
0101b - Alternative 5 (chip-specific).
0110b - Alternative 6 (chip-specific).
0111b - Alternative 7 (chip-specific).
1000b - Alternative 8 (chip-specific).
1001b - Alternative 9 (chip-specific).
1010b - Alternative 10 (chip-specific).
1011b - Alternative 11 (chip-specific).
1100b - Alternative 12 (chip-specific).
1101b - Alternative 13 (chip-specific).
1110b - Alternative 14 (chip-specific).
1111b - Alternative 15 (chip-specific).</td></tr>
<tr><td>7
—</td><td>Reserved</td></tr>
<tr><td>6
DSE</td><td>Drive Strength Enable
Drive strength configuration is valid in all digital pin muxing modes.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>—</td><td>PORTA_PCR17</td></tr><tr><td>—</td><td>PORTB_PCR17</td></tr><tr><td>PORTC_PCR17</td><td>—</td></tr></table>
0b - Low drive strength is configured on the corresponding pin, if pin is configured as a digital output.
1b - High drive strength is configured on the corresponding pin, if pin is configured as a digital output.</td></tr>
<tr><td>5
—</td><td>Reserved</td></tr>
<tr><td>4
—</td><td>Reserved</td></tr>
</table>

Table continues on the next page...

Field	Function
3 —	Reserved
2 SRE	Slew Rate Enable Slew rate configuration is valid in all digital pin muxing modes. 0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output. 1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0b - Internal pull resistor is not enabled on the corresponding pin. 1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

12.3.1.11 Pin Control Register 18 (PCR18)

12.3.1.11.1 Offset

Register	Offset
PCR18	48h

12.3.1.11.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

12.3.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
W								W1C								
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0		MUX				0	DSE	0	PFE	0	SRE	PE	PS
W																
Reset	See Register reset values.															

12.3.1.11.4 Register reset values

Register	Reset value
PCR18	PORTA: 0000_0001h PORTB: 0000_0F17h PORTC: 0000_0001h

12.3.1.11.5 Fields

Field	Function
31 —	Reserved.
30-25 —	Reserved
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected. 1b - 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 —	Reserved.
22-20 —	Reserved
19-16	Interrupt Configuration

Table continues on the next page...

Field	Function
IRQC	The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge. 1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.
15 —	Reserved.
14 —	Reserved.
13-12 —	Reserved
11-8 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: 0000b - Pin disabled (Alternative 0) (analog). 0001b - Alternative 1 (GPIO). 0010b - Alternative 2 (chip-specific). 0011b - Alternative 3 (chip-specific). 0100b - Alternative 4 (chip-specific). 0101b - Alternative 5 (chip-specific). 0110b - Alternative 6 (chip-specific). 0111b - Alternative 7 (chip-specific). 1000b - Alternative 8 (chip-specific). 1001b - Alternative 9 (chip-specific). 1010b - Alternative 10 (chip-specific). 1011b - Alternative 11 (chip-specific). 1100b - Alternative 12 (chip-specific). 1101b - Alternative 13 (chip-specific). 1110b - Alternative 14 (chip-specific). 1111b - Alternative 15 (chip-specific).
7 —	Reserved
6 DSE	Drive Strength Enable Drive strength configuration is valid in all digital pin muxing modes.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>—</td><td>PORTA_PCR18</td></tr><tr><td>—</td><td>PORTB_PCR18</td></tr><tr><td>PORTC_PCR18</td><td>—</td></tr></table>
0b - Low drive strength is configured on the corresponding pin, if pin is configured as a digital output.
1b - High drive strength is configured on the corresponding pin, if pin is configured as a digital output.</td></tr>
<tr><td>5
—</td><td>Reserved</td></tr>
<tr><td>4
PFE</td><td>Passive Filter Enable
Passive filter configuration is valid in all digital pin muxing modes.
NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>—</td><td>PORTA_PCR18</td></tr><tr><td>PORTB_PCR18</td><td>—</td></tr><tr><td>—</td><td>PORTC_PCR18</td></tr></table>
0b - Passive input filter is disabled on the corresponding pin.
1b - 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.</td></tr>
<tr><td>3
—</td><td>Reserved</td></tr>
<tr><td>2
SRE</td><td>Slew Rate Enable
Slew rate configuration is valid in all digital pin muxing modes.
0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.</td></tr>
<tr><td>1
PE</td><td>Pull Enable
Pull configuration is valid in all digital pin muxing modes.
0b - Internal pull resistor is not enabled on the corresponding pin.
1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.</td></tr>
<tr><td>0
PS</td><td>Pull Select
Pull configuration is valid in all digital pin muxing modes.
0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set.
1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.</td></tr>
</table>

12.3.1.12 Pin Control Register 19 (PCR19)

12.3.1.12.1 Offset

Register	Offset
PCR19	4Ch

12.3.1.12.2 Function

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset values for the pins on this device.

See the Chip specific PORT information 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.

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
PORTA _PCR19	—
—	PORTB _PCR19
PORTC _PCR19	—

12.3.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0						ISF	0	0			IRQC			
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	0	0	0		MUX				0	0	0	0	0	SRE	PE	PS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

12.3.1.12.4 Fields

Field	Function
31 —	Reserved.
30-25 —	Reserved
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0b - Configured interrupt is not detected. 1b - 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 —	Reserved.
22-20 —	Reserved
19-16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt, DMA request or trigger as follows: 0000b - Interrupt Status Flag (ISF) is disabled. 0001b - ISF flag and DMA request on rising edge. 0010b - ISF flag and DMA request on falling edge. 0011b - ISF flag and DMA request on either edge. 0100b - Reserved. 0101b - Flag sets on rising edge. 0110b - Flag sets on falling edge. 0111b - Flag sets on either edge. 1000b - ISF flag and Interrupt when logic 0. 1001b - ISF flag and Interrupt on rising-edge. 1010b - ISF flag and Interrupt on falling-edge.

Table continues on the next page...

Field	Function
	1011b - ISF flag and Interrupt on either edge. 1100b - ISF flag and Interrupt when logic 1. 1101b - Enable active high trigger output, flag is disabled. [The trigger output goes to the trigger mux, which allows pins to trigger other peripherals (configurable polarity; 1 pin per port; if multiple pins are configured, then they are ORed together to create the trigger)] 1110b - Enable active low trigger output, flag is disabled. 1111b - Reserved.
15 —	Reserved.
14 —	Reserved.
13-12 —	Reserved
11-8 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: 0000b - Pin disabled (Alternative 0) (analog). 0001b - Alternative 1 (GPIO). 0010b - Alternative 2 (chip-specific). 0011b - Alternative 3 (chip-specific). 0100b - Alternative 4 (chip-specific). 0101b - Alternative 5 (chip-specific). 0110b - Alternative 6 (chip-specific). 0111b - Alternative 7 (chip-specific). 1000b - Alternative 8 (chip-specific). 1001b - Alternative 9 (chip-specific). 1010b - Alternative 10 (chip-specific). 1011b - Alternative 11 (chip-specific). 1100b - Alternative 12 (chip-specific). 1101b - Alternative 13 (chip-specific). 1110b - Alternative 14 (chip-specific). 1111b - Alternative 15 (chip-specific).
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 SRE	Slew Rate Enable Slew rate configuration is valid in all digital pin muxing modes. 0b - Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output.

Table continues on the next page...

Field	Function
	1b - Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0b - Internal pull resistor is not enabled on the corresponding pin. 1b - Internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0b - Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1b - Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

12.3.1.13 Global Pin Control Low Register (GPCLR)

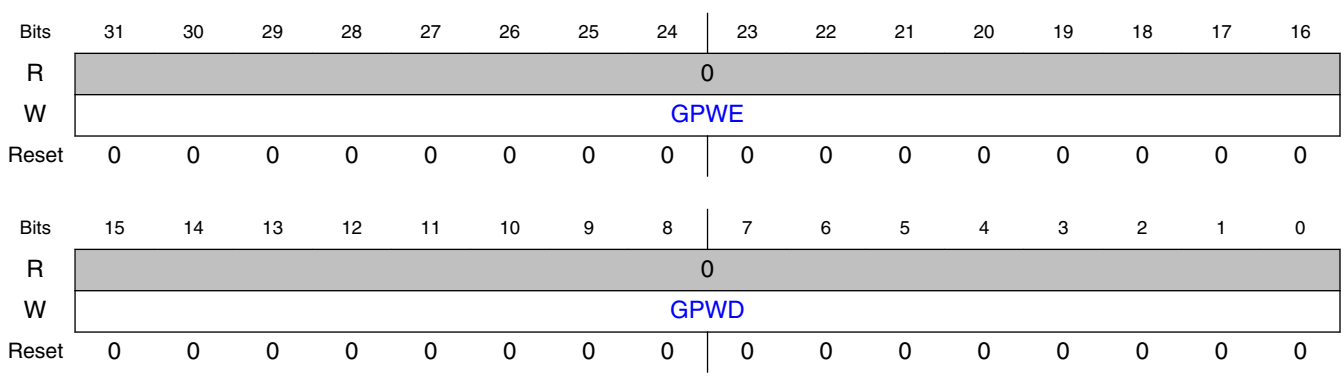
12.3.1.13.1 Offset

Register	Offset
GPCLR	80h

12.3.1.13.2 Function

Only 32-bit writes are supported to this register, any 16-bit or 8-bit writes are ignored.

12.3.1.13.3 Diagram



12.3.1.13.4 Fields

Field	Function
31-16 GPWE	Global Pin Write Enable Selects which Pin Control Registers (15 through 0) bits update with the value in GPWD. 0b - Corresponding lower 16-bits of Pin Control Register is not updated with the value in GPWD. 1b - Corresponding lower 16-bits of Pin Control Register is updated with the value in GPWD.
15-0 GPWD	Global Pin Write Data GPWD field is written to PCRa[15:0] if GPWEn = 1 (a = n).

12.3.1.14 Global Pin Control High Register (GPCHR)

12.3.1.14.1 Offset

Register	Offset
GPCHR	84h

12.3.1.14.2 Function

Only 32-bit writes are supported to this register, any 16-bit or 8-bit writes are ignored.

12.3.1.14.3 Diagram

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

12.3.1.14.4 Fields

Field	Function
31-16 GPWE	Global Pin Write Enable Selects which Pin Control Registers (31 through 16) update with the value in GPWD.

Table continues on the next page...

Field	Function
	0b - Corresponding lower 16-bits of Pin Control Register is not updated with the value in GPWD. 1b - Corresponding lower 16-bits of Pin Control Register is updated with the value in GPWD.
15-0 GPWD	Global Pin Write Data GPWD field is written to PCRa[15:0] if GPWEn = 1 (a = n + 16).

12.3.1.15 Global Interrupt Control Low Register (GICLR)

12.3.1.15.1 Offset

Register	Offset
GICLR	88h

12.3.1.15.2 Function

Only 32-bit writes are supported to this register, any 16-bit or 8-bit writes are ignored.

12.3.1.15.3 Diagram

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

12.3.1.15.4 Fields

Field	Function
31-16 GIWD	Global Interrupt Write Data Write value that is written to upper 16-bits of the Pin Control Registers bits that are selected by GIWE.
15-0 GIWE	Global Interrupt Write Enable Selects which upper 16-bits of Pin Control Registers (15 through 0) update with the value in GIWD. 0b - Corresponding upper 16-bits of Pin Control Register is not updated with the value in GIWD. 1b - Corresponding upper 16-bits of Pin Control Register is updated with the value in GIWD.

12.3.1.16 Global Interrupt Control High Register (GICHR)

12.3.1.16.1 Offset

Register	Offset
GICHR	8Ch

12.3.1.16.2 Function

Only 32-bit writes are supported to this register, any 16-bit or 8-bit writes are ignored.

12.3.1.16.3 Diagram

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

12.3.1.16.4 Fields

Field	Function
31-16 GIWD	Global Interrupt Write Data Write value that is written to upper 16-bits of the Pin Control Registers that are selected by GIWE.
15-0 GIWE	Global Interrupt Write Enable Selects which upper 16-bits of Pin Control Registers (31 through 16) update with the value in GIWD. 0b - Corresponding upper 16-bits of Pin Control Register is not updated with the value in GIWD. 1b - Corresponding upper 16-bits of Pin Register is updated with the value in GIWD.

12.3.1.17 Interrupt Status Flag Register (ISFR)

12.3.1.17.1 Offset

Register	Offset
ISFR	A0h

12.3.1.17.2 Function

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.

12.3.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ISF															
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	ISF															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.3.1.17.4 Fields

Field	Function
31-0 ISF	Interrupt Status Flag Each bit in the field indicates the detection of the configured interrupt of the same number as the field. 0b - Configured interrupt is not detected. 1b - 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.

12.4 Functional description

12.4.1 Pin control

Each port pin has a corresponding Pin Control register, PORT_PCRn, associated with it. The Pin Control register configures the following functions for each pin within the 32-bit port.

- Pullup or pulldown enable
- Slew rate configuration
- 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^2C 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.

12.4.2 Global pin control

The two global pin control registers allow a single register write to update the lower 16-bits of the Pin Control Register for 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 same port with the same peripheral function.

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

12.4.3 Global interrupt control

The two global interrupt control registers allow a single register write to update the upper 16-bits of the Pin Control Register for up to 16 pins, all with the same value.

The global interrupt control registers are designed to enable software to quickly configure multiple pins within the same port with the same interrupt configuration.

The global interrupt control registers are write-only registers and always read as 0.

12.4.4 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
- Rising edge flag (for software polling)
- Falling edge flag (for software polling)
- Rising and falling edge flag (for software polling)
- Active high level peripheral trigger (status flag disabled)
- Active low level peripheral trigger (status flag disabled)
- 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 pin 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.

The PORT module generates a single peripheral trigger output that asserts if any pin configured for active high trigger is logic one, or any pin triggered for active low trigger is logic zero. The peripheral trigger output asynchronously updates from the value on the configured pins.

Chapter 13
System Integration Module (SIM)

13.1 Introduction

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.

13.1.1 Features

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

NOTE

See chip-specific section LPUART instantiation information to know the number of LPUART instances supported by this device.

13.2 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

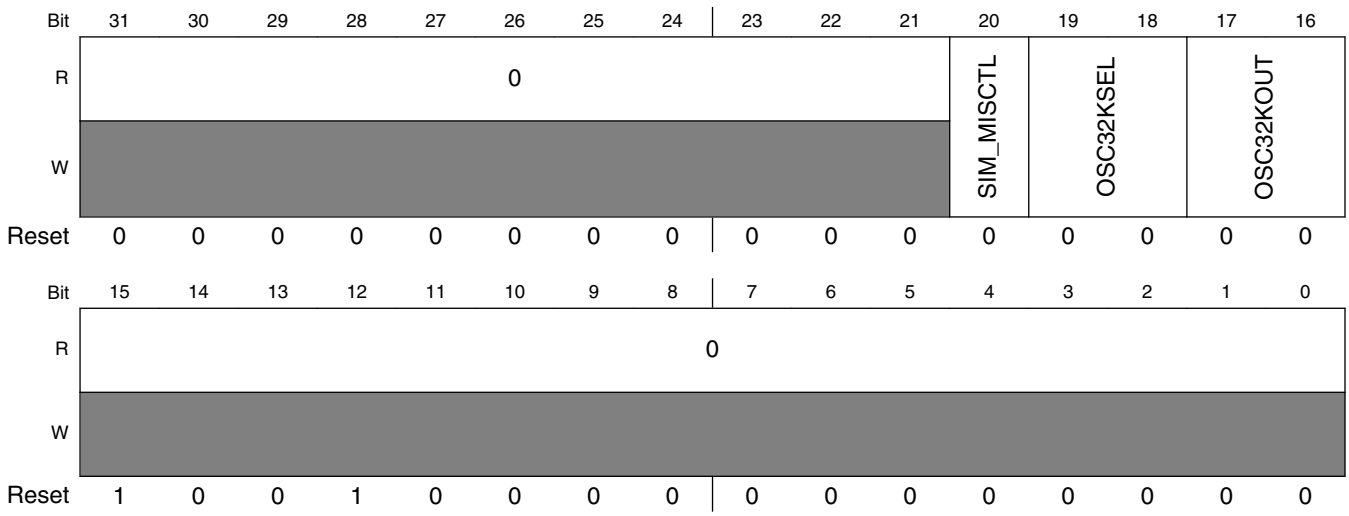
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_7000	System Options Register 1 (SIM_SOPT1)	32	R/W	0000_9000h	13.2.1/315
4004_8004	System Options Register 2 (SIM_SOPT2)	32	R/W	0000_0000h	13.2.2/316
4004_800C	System Options Register 4 (SIM_SOPT4)	32	R/W	0000_0000h	13.2.3/318
4004_8010	System Options Register 5 (SIM_SOPT5)	32	R/W	0000_0000h	13.2.4/319
4004_8018	System Options Register 7 (SIM_SOPT7)	32	R/W	0000_0000h	13.2.5/321
4004_8024	System Device Identification Register (SIM_SDID)	32	R	See section	13.2.6/322
4004_8034	System Clock Gating Control Register 4 (SIM_SCGC4)	32	R/W	F000_0030h	13.2.7/324
4004_8038	System Clock Gating Control Register 5 (SIM_SCGC5)	32	R/W	0200_0182h	13.2.8/326
4004_803C	System Clock Gating Control Register 6 (SIM_SCGC6)	32	R/W	0000_0001h	13.2.9/328
4004_8040	System Clock Gating Control Register 7 (SIM_SCGC7)	32	R/W	0000_0100h	13.2.10/331
4004_8044	System Clock Divider Register 1 (SIM_CLKDIV1)	32	R/W	See section	13.2.11/331
4004_804C	Flash Configuration Register 1 (SIM_FCFG1)	32	R/W	See section	13.2.12/333
4004_8050	Flash Configuration Register 2 (SIM_FCFG2)	32	R	See section	13.2.13/336
4004_8058	Unique Identification Register Mid-High (SIM_UIDMH)	32	R	See section	13.2.14/337
4004_805C	Unique Identification Register Mid Low (SIM_UIDML)	32	R	See section	13.2.15/338
4004_8060	Unique Identification Register Low (SIM_UIDL)	32	R	See section	13.2.16/338
4004_8100	COP Control Register (SIM_COPC)	32	R/W	0000_000Ch	13.2.17/339
4004_8104	Service COP (SIM_SRVCOP)	32	W	0000_0000h	13.2.18/340

13.2.1 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



SIM_SOPT1 field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 SIM_MISCTL	This bit controls the function of BLE_RF_ACTIVE on PTC1/PTC19 ALT7 0 Chip low power mode output to PAD 1 BLE active output to PAD
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.

Table continues on the next page...

SIM_SOPT1 field descriptions (continued)

Field	Description
	10 Reserved. 11 Reserved.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

13.2.2 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		LPUART1SRC		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 LPUART1SRC	LPUART1 Clock Source Select Selects the clock source for the LPUART1 transmit and receive clock. 00 Clock disabled 01 MCGFLLCLK clock 10 OSCERCLK clock 11 MCGIRCLK clock
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

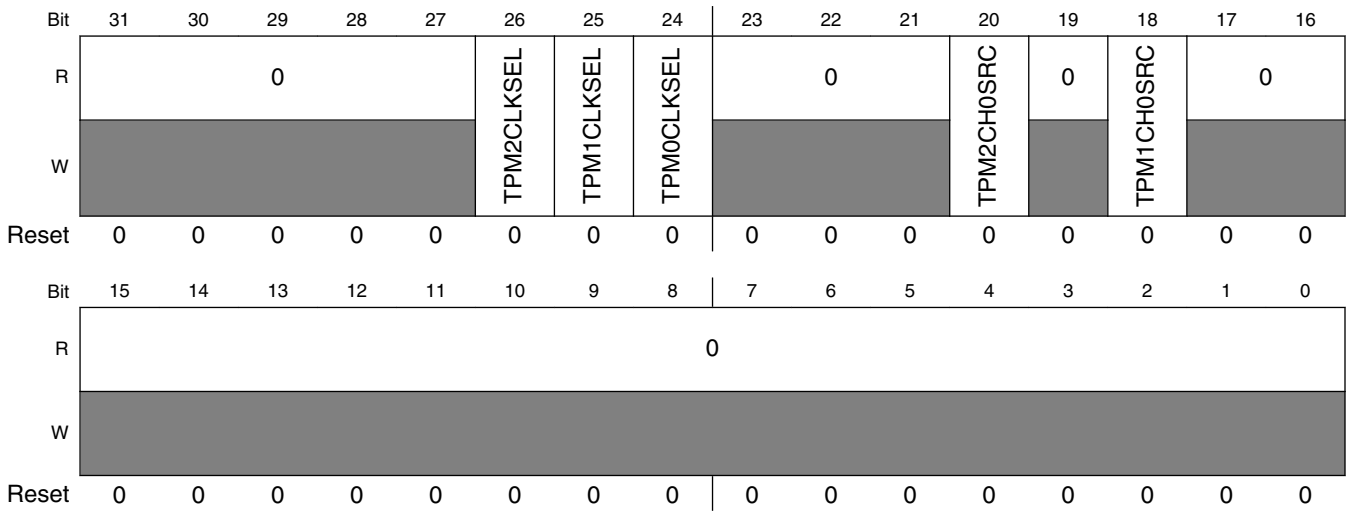
Table continues on the next page...

SIM_SOPT2 field descriptions (continued)

Field	Description
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.
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.

13.2.3 System Options Register 4 (SIM_SOPT4)

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



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. **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.

Table continues on the next page...

SIM_SOPT4 field descriptions (continued)

Field	Description
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.

13.2.4 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	LPUART1ODE	LPUART0ODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

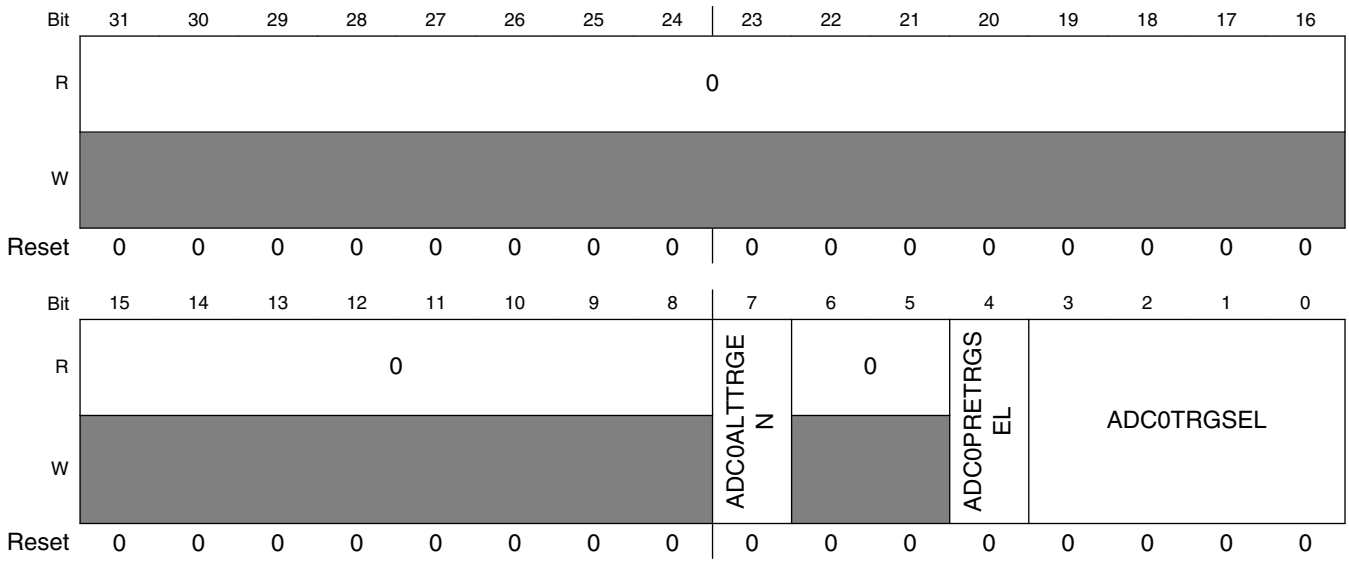
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									LPUART1RXS RC	LPUART1TXS RC		0	LPUART0RXS RC	LPUART0TXS RC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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 LPUART1ODE	LPUART1 Open Drain Enable 0 Open drain is disabled on LPUART1. 1 Open drain is enabled on LPUART1
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 LPUART1RXSRC	LPUART1 Receive Data Source Select Selects the source for the LPUART1 receive data. 0 LPUART1_RX pin 1 CMP0 output
5–4 LPUART1TXSRC	LPUART1 Transmit Data Source Select Selects the source for the LPUART1 transmit data. 00 LPUART1_TX pin 01 LPUART1_TX pin modulated with TPM1 channel 0 output 10 LPUART1_TX pin modulated with TPM2 channel 0 output 11 Reserved
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 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

13.2.5 System Options Register 7 (SIM_SOPT7)

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



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.
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.

Table continues on the next page...

SIM_SOPT7 field descriptions (continued)

Field	Description
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 interrupt and RTC seconds interrupt 1101 RTC seconds 1110 LPTMR0 trigger 1111 Radio TSM

13.2.6 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	1	0	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.
- SRAMSIZE 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. 0011 KW3x Family (BTLE)
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 SUBFAMID	Kinetis Sub-Family ID. Specifies the Kinetis sub-family of the device. 101 KWx5 Sub family 110 KWx6 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 0111 64 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 Reserved 0011 40-pin 0100 48-pin 0101 Reserved 0110 Reserved 0111 Reserved 1000 Reserved 1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved

Table continues on the next page...

SIM_SDID field descriptions (continued)

Field	Description
	1110 Reserved 1111 Reserved

13.2.7 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	VREF	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 VREF	VREF Clock Gate Control Controls the clock gate to the VREF module. 0 Clock disabled 1 Clock enabled
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.

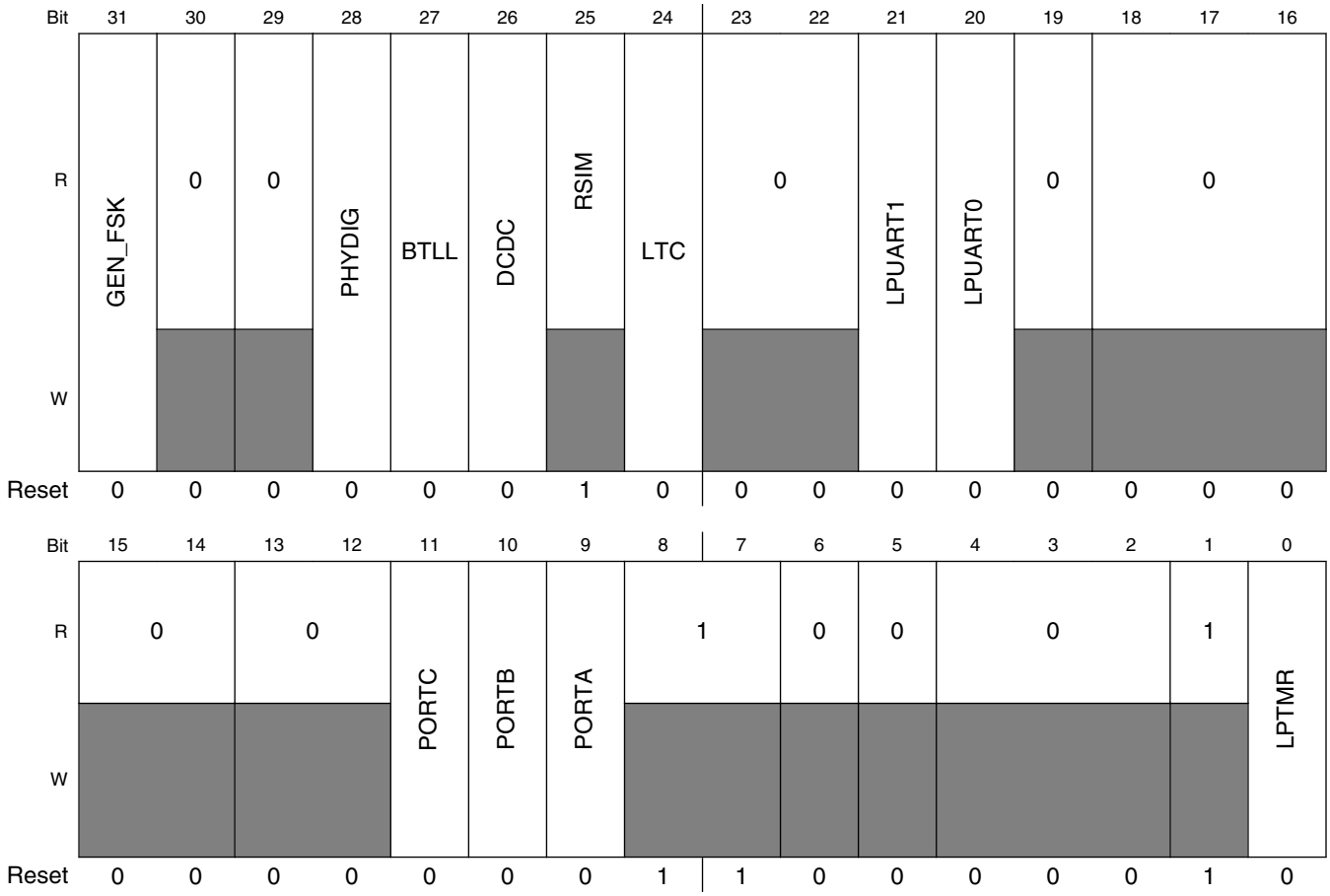
Table continues on the next page...

SIM_SCGC4 field descriptions (continued)

Field	Description
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^2C1 module. 0 Clock disabled 1 Clock enabled
6 I2C0	I2C0 Clock Gate Control Controls the clock gate to the I^2C0 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.
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.

13.2.8 System Clock Gating Control Register 5 (SIM_SCGC5)

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



SIM_SCGC5 field descriptions

Field	Description
31 GEN_FSK	Generic FSK enabled 0 GFSK CGC bit disabled. 1 GFSK CGC bit enabled.
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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

Table continues on the next page...

SIM_SCGC5 field descriptions (continued)

Field	Description
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 LPUART1	LPUART1 Clock Gate Control This bit controls the clock gate to the LPUART1 module. 0 Clock disabled 1 Clock enabled
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.
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.

Table continues on the next page...

SIM_SCGC5 field descriptions (continued)

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

13.2.9 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	0	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				FLEXCAN0	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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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
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.

Table continues on the next page...

SIM_SCGC6 field descriptions (continued)

Field	Description
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
8–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 FLEXCAN0	FLEXCAN0 Clock Gate Control This bit controls the clock gate to the FLEXCAN0 module. 0 Clock disabled 1 Clock enabled
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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

13.2.10 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.

13.2.11 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 FTFE_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:

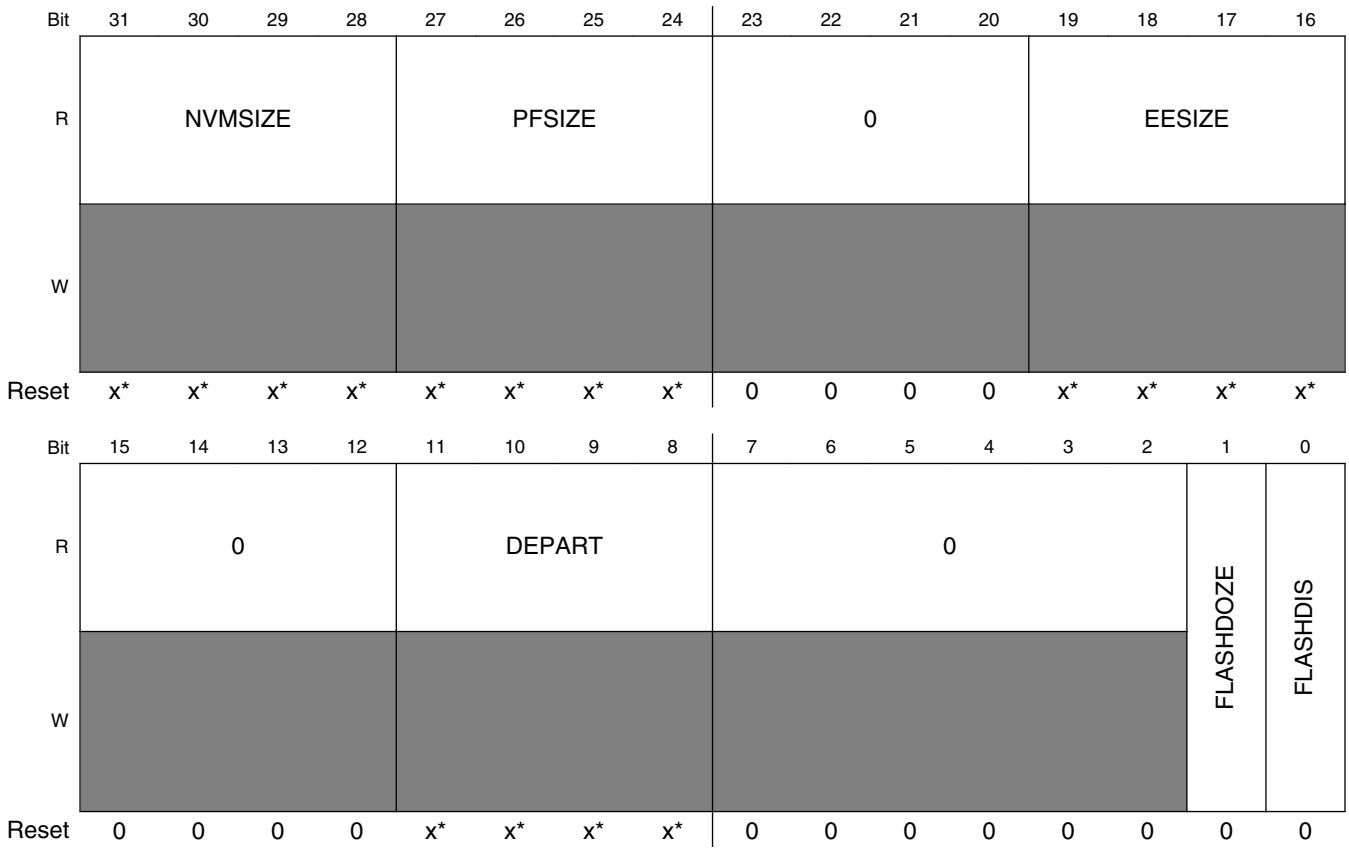
- OUTDIV1 field: The reset value depends on the FTFE_FOPT[LPBOOT]. It is loaded with 0000 (divide by 1), 0001 (divide by 2), 0011 (divide by 4, or 0111 (divide by 8).

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 FTFE_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.
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.

13.2.12 Flash Configuration Register 1 (SIM_FCFG1)

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



* Notes:
- x = Undefined at reset.

SIM_FCFG1 field descriptions

Field	Description
31–28 NVMSIZE	FlexNVM Size Specifies the amount of FlexNVM memory available on the device. 0000 0 KB 0001 Reserved 0010 Reserved 0011 Reserved 0100 Reserved 0101 Reserved 0110 Reserved 0111 Reserved 1000 Reserved 1001 256 KB, 16 KB protection region 1010 Reserved

Table continues on the next page...

SIM_FCFG1 field descriptions (continued)

Field	Description
	1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 256 KB, 16 KB protection region
27–24 PFSIZE	Program Flash Size Specifies the amount of program flash memory available on the device . Undefined values are reserved. 1001 256 KB of program flash memory 1011 512 KB of program flash memory 1111 Reserved
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 EESIZE	EEPROM Size Specifies the EEPROM data size. 0000 Reserved 0001 8 KB 0010 4 KB 0011 2 KB 0100 1 KB 0101 512 bytes 0110 256 bytes 0111 128 bytes 1000 64 bytes 1001 32 bytes 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 DEPART	FlexNVM partition For devices with FlexNVM: Data flash/EEPROM backup split. See DEPART bit description in FTFE chapter. For devices without FlexNVM: Reserved
7–2 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

Table continues on the next page...

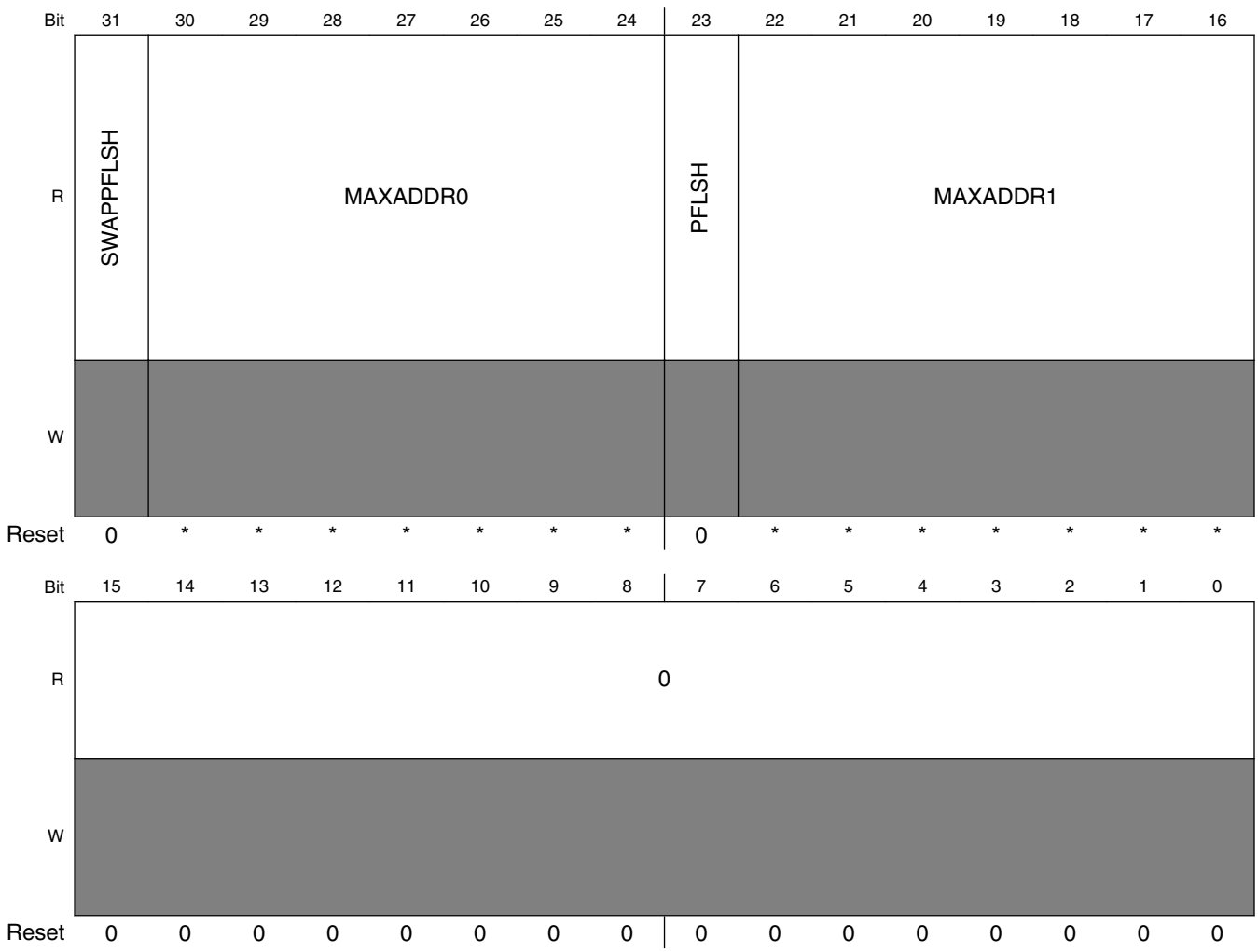
SIM_FCFG1 field descriptions (continued)

Field	Description
	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.

13.2.13 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



* 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 SWAPPFLSH	Swap program flash For devices without FlexNVM: Indicates that swap is active.

Table continues on the next page...

SIM_FCFG2 field descriptions (continued)

Field	Description
	0 Swap is not active. 1 Swap is active.
30–24 MAXADDR0	Max Address block 0 This field concatenated with 13 trailing zeros indicates the first invalid address of program flash (block 0). For example, if MAXADDR0 = 0x20, the first invalid address of program flash (block 0) is 0x0004_0000. This would be the MAXADDR0 value for a device with 256 KB program flash in flash block 0.
23 PFLSH	Program flash only For devices with FlexNVM, this bit is always clear. For devices without FlexNVM, this bit is always set. 0 Device supports FlexNVM. 1 Program Flash only, device does not support FlexNVM.
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 = 0x20 the first invalid address of flash block 1 is 0x4_0000 + 0x4_0000. This would be the MAXADDR1 value for a device with 512 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.

13.2.14 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	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

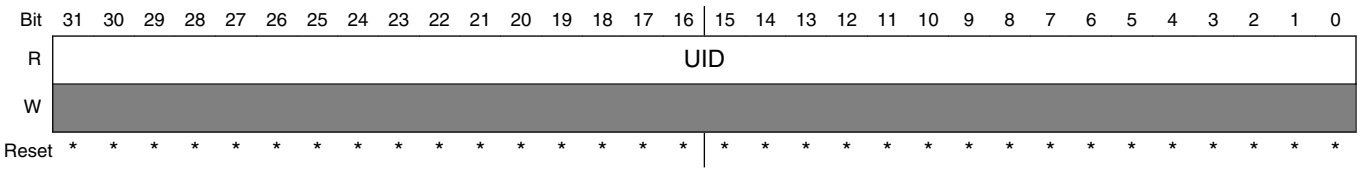
* Notes:
- UID field: Device specific value.

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.

13.2.15 Unique Identification Register Mid Low (SIM_UIDML)

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



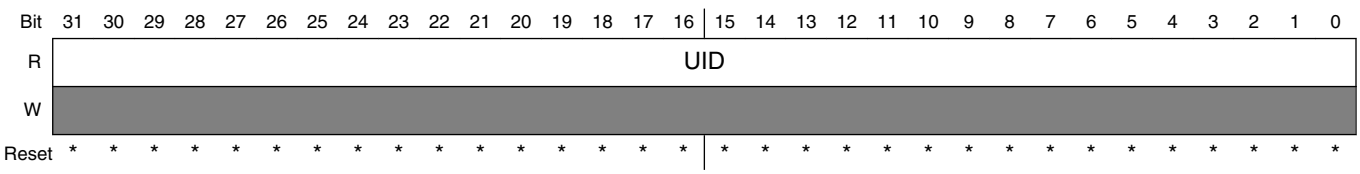
* Notes:
- UID field: Device specific value.

SIM_UIDML field descriptions

Field	Description
UID	Unique Identification Unique identification for the device.

13.2.16 Unique Identification Register Low (SIM_UIDL)

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



* Notes:
- UID field: Device specific value.

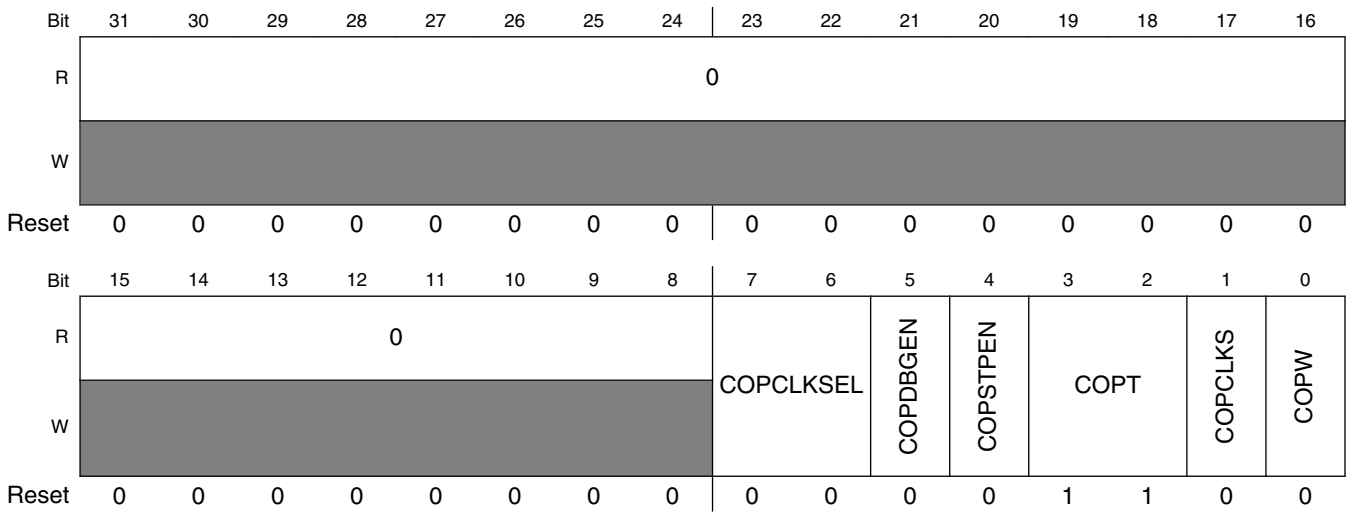
SIM_UIDL field descriptions

Field	Description
UID	Unique Identification Unique identification for the device.

13.2.17 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



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

Table continues on the next page...

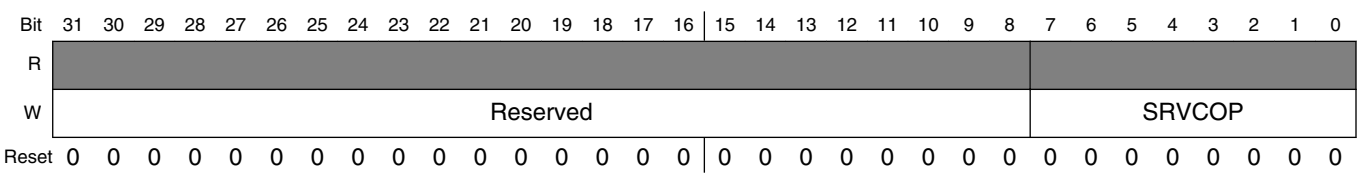
SIM_COPC field descriptions (continued)

Field	Description
	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

13.2.18 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



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 14
System Mode Controller (SMC)

14.1 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.

14.2 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 Kinetis microcontrollers.

The following table shows the translation between the ARM CPU modes and the Kinetis 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 Kinetis MCU documentation normally uses wait and stop.

In addition, Kinetis 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 14-1. 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 14-1. 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 the voltage to internal logic and powering down the system RAM3 partition. The system RAM2 partition can be optionally retained using STOPCTRL[RAM2PO]. 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 RAM3 partition. The system RAM2 partition can be optionally retained using STOPCTRL[RAM2PO].The system RAM1 partition contents are retained in this mode. Internal logic states are not retained. [1]
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.

14.3 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	14.3.1/344
4007_E001	Power Mode Control register (SMC_PMCTRL)	8	R/W	00h	14.3.2/345
4007_E002	Stop Control Register (SMC_STOPCTRL)	8	R/W	03h	14.3.3/346
4007_E003	Power Mode Status register (SMC_PMSTAT)	8	R	01h	14.3.4/348

14.3.1 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.

14.3.2 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. Software should write 0 to this bit to maintain compatibility.
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 successful. 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

14.3.3 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 Write	PSTOPO		PORPO	RAM2PO	Reserved	LLSM		
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 RAM2PO	RAM2 Power Option This bit controls powering of RAM partition 2 in LLS2 or VLLS2 mode. **NOTE:** See the device's Chip Configuration details for the size and location of RAM partition 2 0 RAM2 not powered in LLS2/VLLS2 1 RAM2 powered in LLS2/VLLS2
3 Reserved	This field is reserved. This bit is reserved for future expansion. Software should write 0 to this bit to maintain compatibility.
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

Table continues on the next page...

SMC_STOPCTRL field descriptions (continued)

Field	Description
	101 Reserved 110 Reserved 111 Reserved

14.3.4 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

14.4 Functional description

14.4.1 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.

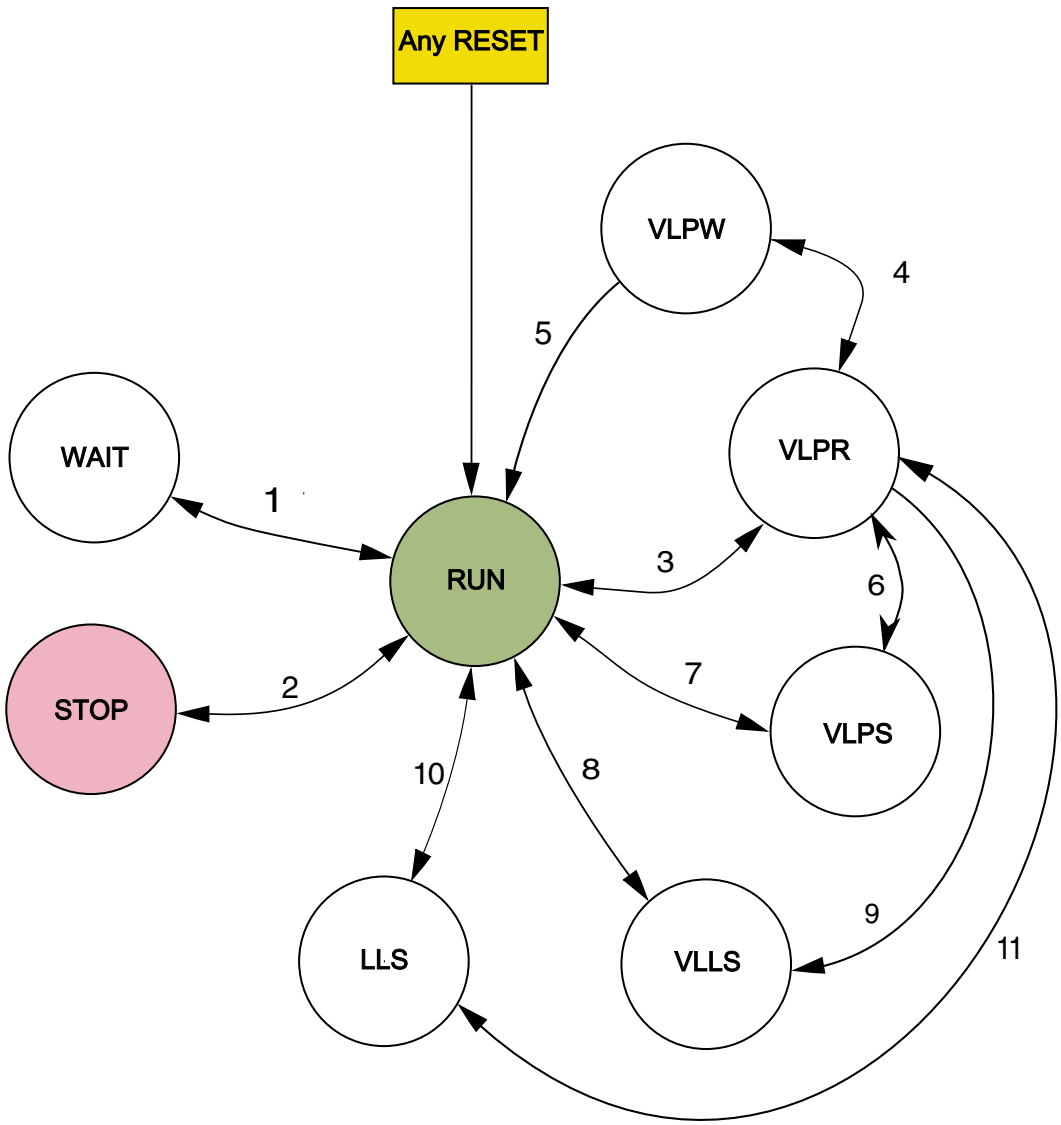


Figure 14-1. Power mode state diagram

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

Table 14-2. 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.[1]
	WAIT	RUN	Interrupt or Reset
2	RUN	STOP	PMCTRL[RUNM]=00, PMCTRL[STOPM]=000[2] 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

Table continues on the next page...

Table 14-2. Power mode transition triggers (continued)

Transition #	From	To	Trigger conditions
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.
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 and LLSx mode was entered directly from RUN 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.
	LLSx	VLPR	Wakeup from enabled LLWU input source and LLSx mode was entered directly from VLPR **NOTE:** If LLSx was entered directly from RUN, hardware will not allow this transition and will force exit back to RUN

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

14.4.2 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.

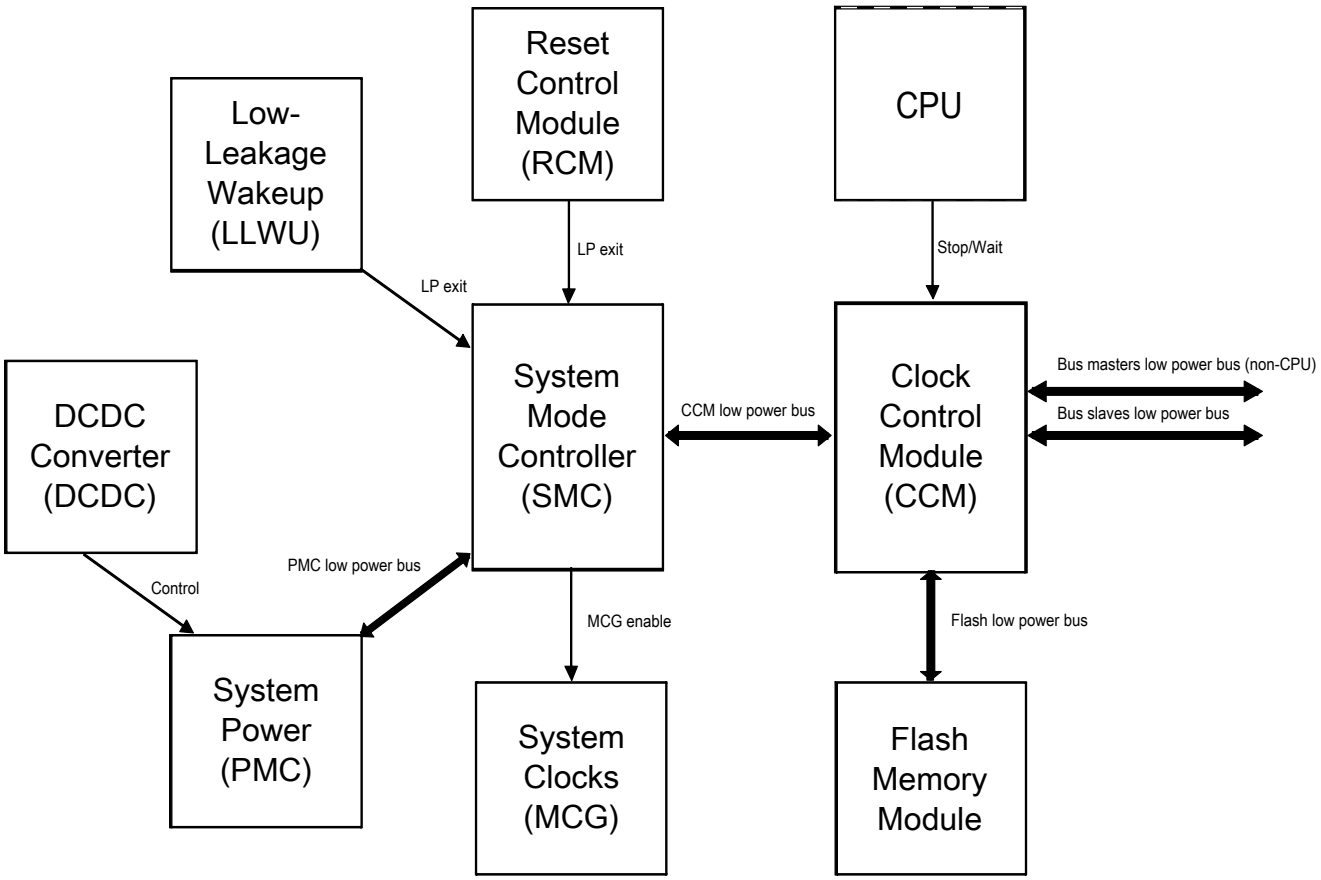


Figure 14-2. Low-power system components and connections

14.4.2.1 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.

14.4.2.2 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.

14.4.2.3 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.

14.4.2.4 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.

14.4.2.5 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.

14.4.3 Run modes

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

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

14.4.3.1 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.

14.4.3.2 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.

14.4.4 Wait modes

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

- Wait
- Very-Low Power Wait (VLPW)

14.4.4.1 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.

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 causes an exit from WAIT mode, returning the device to normal RUN mode.

14.4.4.2 Very-Low-Power Wait (VLPW) mode

VLPW mode is entered by entering the Sleep-Now or Sleep-On-Exit mode while SLEEPDEEP is cleared and the device 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 device at a reduced frequency. To further reduce power in this mode, disable the clocks to unused modules.

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 causes an exit from VLPW mode, returning the device to normal RUN mode.

14.4.5 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.

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)

14.4.5.1 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.

14.4.5.2 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.

14.4.5.3 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 14-2.

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 the run mode from which LLS was entered (either normal RUN or VLPR) 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.

14.4.5.4 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 14-2.

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 RESET pin will cause an exit from any VLLS mode, returning the device to normal RUN mode. When exiting VLLS via the RESET pin, RCM_SRS[PIN] and RCM_SRS[WAKEUP] are set.

14.4.6 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 15
Power Management Controller (PMC)

15.1 Introduction

The power management controller (PMC) contains the internal voltage regulator, power on reset (POR), and low voltage detect system (LVD) for the VDD domain.

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

15.2 Features

A list of included PMC features can be found here.

- Internal voltage regulator
- Active POR providing brown-out detect
- Low-voltage detect (LVD) on VDD supporting two low-voltage trip points with four warning levels per trip point

15.3 Low-voltage detect (LVD) system

This device includes a system to guard against low-voltage conditions on the VDD supply. When the VDD supply falls below a specific trip point, the LVD circuit puts the device into a reset state, preventing the device from attempting to operate below its operating voltage range.

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.

15.3.1 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.

15.3.2 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].

15.3.3 Low-voltage warning (LVW) interrupt operation

The LVD system that monitors the VDD supply 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}

15.4 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.

15.5 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	15.5.1/364
4007_D001	Low Voltage Detect Status And Control 2 register (PMC_LVDSC2)	8	R/W	00h	15.5.2/365
4007_D002	Regulator Status And Control register (PMC_REGSC)	8	R/W	04h	15.5.3/366

15.5.1 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.

Table continues on the next page...

PMC_LVDSC1 field descriptions (continued)

Field	Description
	0 Low-voltage event not detected 1 Low-voltage event detected
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

15.5.2 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 ($V_{LVW} = V_{LVW4}$)

15.5.3 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	BGEN	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 BGEN	Bandgap Enable In VLPx Operation BGEN controls whether the bandgap is enabled in lower power modes of operation (VLPx, LLS, and VLLSx). When on-chip peripherals require the bandgap voltage reference in low power modes of operation, set BGEN to continue to enable the bandgap operation. **NOTE:** When the bandgap voltage reference is not needed in low power modes, clear BGEN to avoid excess power consumption. 0 Bandgap voltage reference is disabled in VLPx , LLS , and VLLSx modes. 1 Bandgap voltage reference is enabled in VLPx , LLS , and VLLSx modes.
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.

Table continues on the next page...

PMC_REGSC field descriptions (continued)

Field	Description
	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 Enables the bandgap buffer. 0 Bandgap buffer not enabled 1 Bandgap buffer enabled

Chapter 16
DCDC Converter (DCDC)

16.1 Introduction

16.2 Features

The DCDC module includes the following features:

- Single inductor, multiple outputs.
- Buck mode (pin selectable: CFG = DCDC_IN → buck;). This is only configurable through hardware.
- Continuous or pulsed operation (software configurable).
- Power switch input pin to allow external control of power up in buck manual start mode.
- Scaled battery output voltage suitable for analog-to-digital converter (ADC) utilization.
- Internal oscillator for support where the crystal oscillator is not present.

16.3 Block diagram

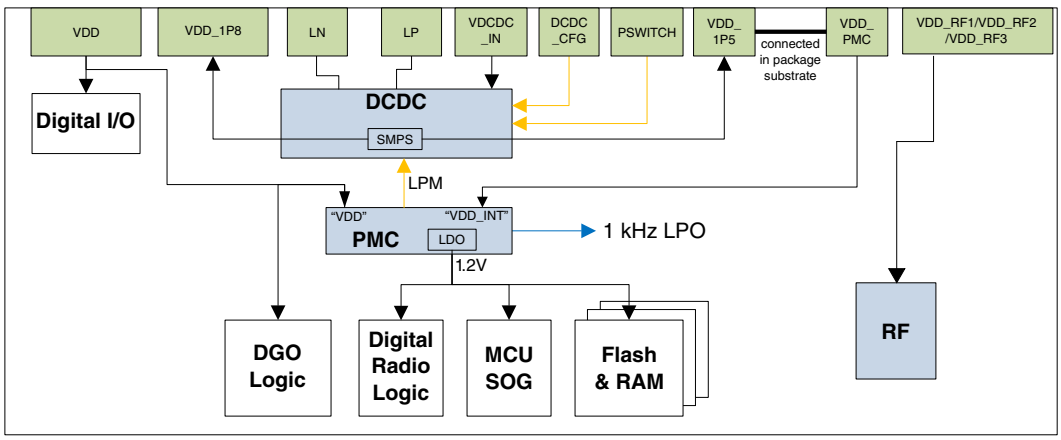


Figure 16-1. DCDC block Diagram

16.4 Functional Description

The DCDC converter module is a switching mode DCDC converter supporting Buck, 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 Buck mode . They are VDD_1P5 and VDD_1P8.

Selection of operating mode is done by setting the pin DCDC_CFG. VDD_1P5 regulated output supplies the radio block and the SOG (Sea of Gates). VDD_1P8 regulated output supplies the rest of the device. VDD_1P8 output can feed external circuitry. Refer to the part specific datasheet for current limits on VDD_1P8.

PSWITCH pin can be used to wakeup the DCDC converter in buck mode. When the PSWITCH voltage (with respect to GND) reaches the threshold voltage (V_{IH} with respect to VDCDC_IN), the DCDC will begin regulation. After the DCDC turn-on time has expired ($T_{DCDC\_ON\_BUCK}$), the PSWITCH voltage may be removed if manual mode operation is desired. Note that the application design must ensure that VDCDC_IN comes up before the PSWITCH voltage and the PSWITCH voltage must not exceed VDCDC_IN.

16.5 Application Requirements

Continuous mode

In continuous mode, the control loop of the DCDC converter is always outputting a PWM signal which charges the tank capacitor to the target voltage. Continuous mode is the default setting for the DCDC and the DCDC is always in continuous mode when the device is in normal RUN, or HSRUN (if HSRUN is available). Continuous mode is optionally available in VLPR, VLPW, or VLPS and never available for LLS mode, or VLLSx modes.

The following register settings are required to ensure optimum DCDC operation in continuous mode:

- DCDC_REG1[DCDC_LOOPCTRL_EN_DF_HYST] = 1
- DCDC_REG1[DCDC_LOOPCTRL_EN_CM_HYST] = 1
- DCDC_REG2[DCDC_LOOPCTRL_HYST_SIGN] = 1

Target voltage adjustment

To adjust target voltage of VDD_1P8 and VDD_1P5 the following steps must be followed:

1. Clear DCDC_VDD1P8CTRL_DISABLE_STEP and DCDC_VDD1P5CTRL_DISABLE_STEP
2. Set the target register bits, DCDC_VDD1P5CTRL_TRG_BUCK and DCDC_VDD1P8CTRL_TRG as per the bitfield values.

DCDC_STS_DC_OK bit will be de-asserted after target register changes. After the output voltage settles to the new target value, DCDC_STS_DC_OK will be asserted.

NOTE

DCDC output voltage can be modified only in continuous mode.

Pulsed mode

In pulsed mode, the control loop of the DCDC converter only turns on the PWM signal which charges the tank capacitor when the output voltage drops below a specified level (defined by the hysteresis level specified in DCDC_REG0[DCDC_LP_STATE_HYS_L]). The PWM charging signal will then turn off once the output voltage has reached the target voltage (defined by the hysteresis level

specified in DCDC_REG0[DCDC_LP_STATE_HYS_H]). Pulse mode is only available in low power modes (VLPR, VLPW, VLPS, LLS, or VLLSx modes) and will only be effective for light loads on the DCDC (<= 0.5 mA).

Before entering pulsed mode, must set DCDC_VDD1P8CTRL_DISABLE_STEP and DCDC_VDD1P5CTRL_DISABLE_STEP bit to 1.

The following register settings are required to ensure optimum DCDC operation:

- DCDC_REG1[DCDC_LOOPCTRL_EN_DF_HYST] = 1
- DCDC_REG1[DCDC_LOOPCTRL_EN_CM_HYST] = 1
- DCDC_REG2[DCDC_LOOPCTRL_HYST_SIGN] = 1
- DCDC_REG0[DCDC_LP_DF_CMP_ENABLE] = 1

Half/Double FET

When current loading is less than the half FET threshold current, it is recommended to use the half FET mode which can improve efficiency. When current loading is greater than the double FET threshold current, it is recommended to use the double FET mode which can improve the current drive capability and efficiency.

16.6 Assumptions

- PSWITCH cannot be shorted to DCDC_IN in Lithium-ion battery configurations.
- PSWITCH can be shorted to battery (except Lithium-ion battery) for automatic and continuous turn on.
- DCDC remains ON when PSWITCH is connected to VDD. After the DCDC ON time (T_{DCDC_ON}), the PSWITCH state may switch to the low state and the DCDC will remain ON. Software will then be able to shut the DCDC OFF, when PWSITCH is in low state.
- PSWITCH must be asserted until DCDC control stabilizes. If not, DCDC shuts down.
- DCDC receives signal from PMC LVD indicating 1.8 V is good.
- DCDC, when starting, initializes both 1P5 and 1P8 and then the main loop starts. After the DCDC_OK_STATUS bit is asserted, software can change the configuration of the DCDC to the required configuration, which is based on the battery voltage and desired IO configuration.
- DCDC outputs signal that DCDC voltages are above target values.

16.7 Memory map and register definition

16.7.1 DCDC Register Descriptions

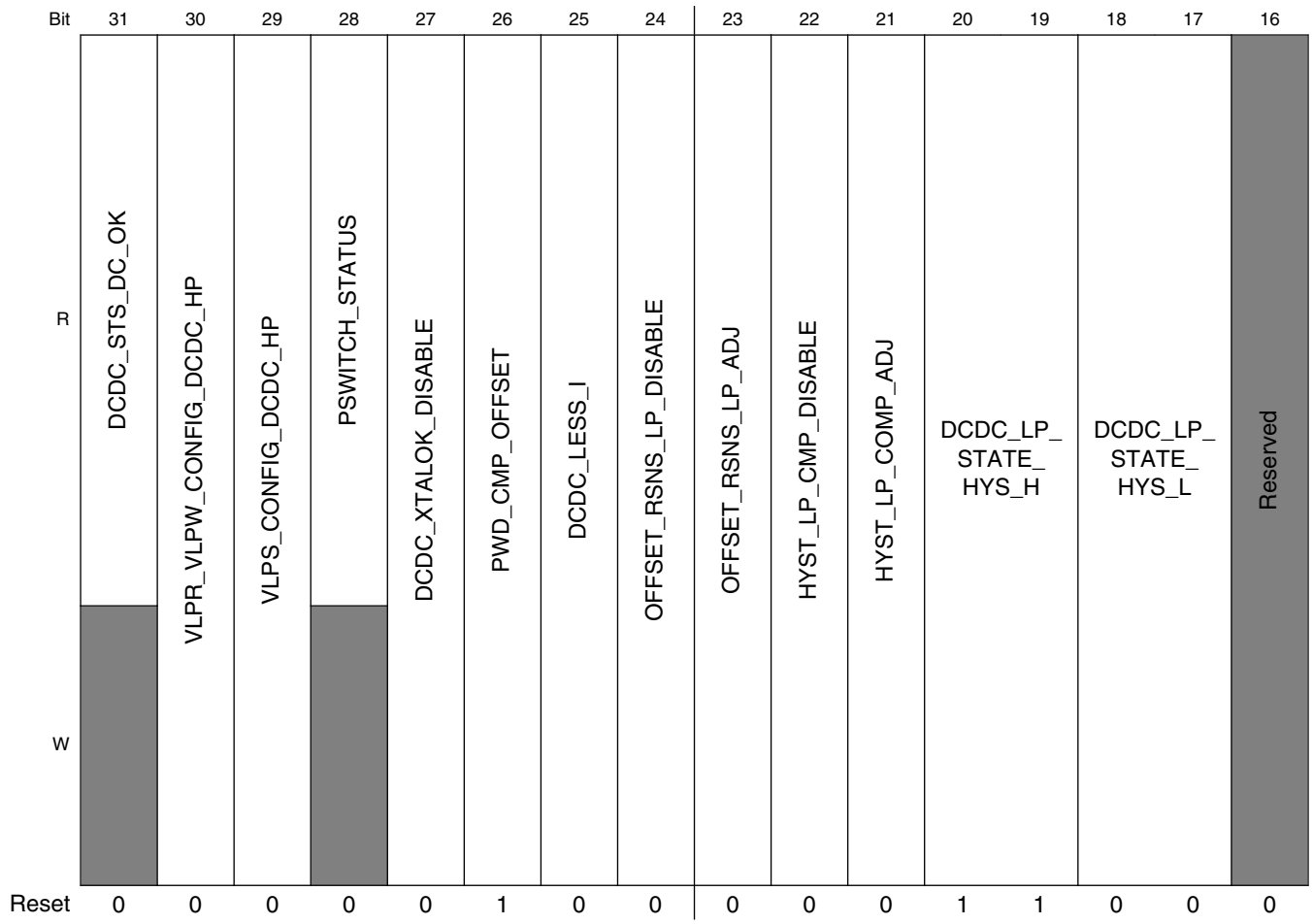
All the registers will only be reset to default value after POR reset.

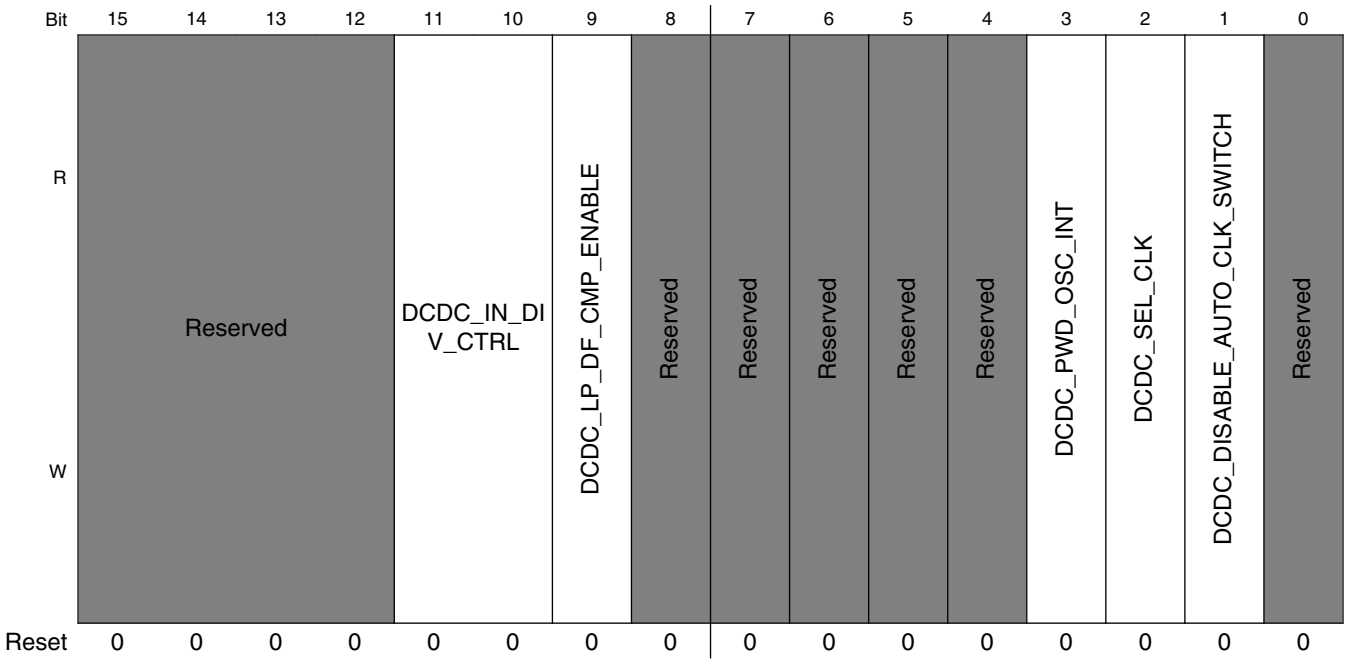
DCDC memory map

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	16.7.2/374
4005_A004	DCDC REGISTER 1 (DCDC_REG1)	32	R/W	0017_C21Ch	16.7.3/378
4005_A008	DCDC REGISTER 2 (DCDC_REG2)	32	R/W	0000_4009h	16.7.4/379
4005_A00C	DCDC REGISTER 3 (DCDC_REG3)	32	R/W	0000_AA46h	16.7.5/381
4005_A010	DCDC REGISTER 4 (DCDC_REG4)	32	R/W	0000_0000h	16.7.6/385
4005_A018	DCDC REGISTER 6 (DCDC_REG6)	32	R/W	0000_0000h	16.7.7/386
4005_A01C	DCDC REGISTER 7 (DCDC_REG7)	32	R/W	0000_0000h	16.7.8/387

16.7.2 DCDC REGISTER 0 (DCDC_REG0)

Address: 4005_A000h base + 0h offset = 4005_A000h





DCDC_REG0 field descriptions

Field	Description
31 DCDC_STS_DC_OK	Status bit to indicate that the DCDC output voltage is stable. The lock time depends on the loading and the DCDC mode. When changing output voltage target, it will take approximately (0.5 ms* target change steps). In pulsed mode, it will take approximately 5 ms. In startup, it will take approximately 50 ms. 0 Unstable DCDC output voltage 1 Stable DCDC output voltage
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. **NOTE:** This bit should only be changed in RUN mode. 0 DCDC works in pulsed mode when SoC is in VLPR / VLPW modes. 1 DCDC works in continuous 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. 0 DCDC works in pulsed mode when SOC is in VLPS modes. 1 DCDC works in continuous mode when SOC is in VLPS modes.
28 PSWITCH_STATUS	Status bit to indicate PSWITCH status 0 PSWITCH is low 1 PSWITCH is high
27 DCDC_XTALOK_DISABLE	Disable xtalok detection circuit. This is a debug bit which should not be changed in real case. This bit disables the clock 24 MHz reference clock detector.
26 PWD_CMP_OFFSET	Output range comparator monitors the output voltage of DCDC. When the DCDC output voltage is out of range, this comparator will speed up DCDC control loop in an attempt to bring the DCDC output voltage back in range.

Table continues on the next page...

DCDC_REG0 field descriptions (continued)

Field	Description
	0 Output range comparator powered up. 1 Output range comparator powered down.
25 DCDC_LESS_I	Reduces DCDC current by reducing the analog reference current inside the DCDC Converter. As a result, you may see higher ripple amplitude. **NOTE:** This bit only affects RUN mode. 0 Use normal current for analog references 1 Use reduced current for analog references
24 OFFSET_RSNS_ LP_DISABLE	Disable hysteresis in low power voltage sense. **NOTE:** It is recommended to always ensure that this bit is written with its reset value. 0 Hysteresis feature is enabled 1 Hysteresis feature is disabled
23 OFFSET_RSNS_ LP_ADJ	Adjust hysteretic value in low power voltage sense. **NOTE:** It is recommended to always ensure that this bit is written with its reset value. 0 Adjustment feature is disabled 1 Adjustment feature is enabled
22 HYST_LP_CMP_ DISABLE	Disable hysteresis in low power comparator. **NOTE:** It is recommended to always ensure that this bit is written with its reset value. 0 Hysteresis feature is enabled 1 Hysteresis feature is disabled
21 HYST_LP_ COMP_ADJ	Adjust hysteretic value in low power comparator. **NOTE:** It is recommended to always ensure that this bit is written with its reset value. 0 Adjustment feature is disabled 1 Adjustment feature is enabled
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

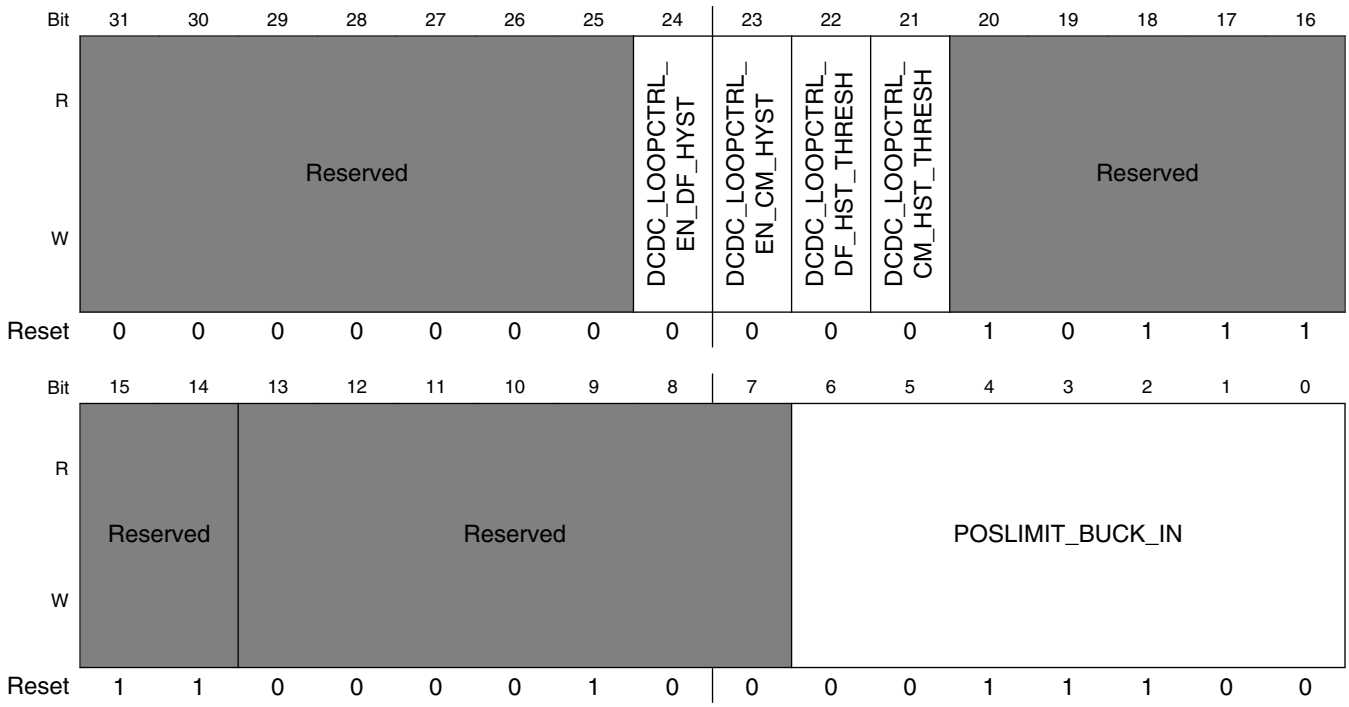
Table continues on the next page...

DCDC_REG0 field descriptions (continued)

Field	Description
11–10 DCDC_IN_DIV_CTRL	Controls DCDC_IN voltage divider. The divided DCDC_IN output is input to an ADC channel which allows the battery voltage to be measured. 00 OFF 01 DCDC_IN 10 DCDC_IN / 2 11 DCDC_IN / 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 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. This is the recommended configuration to guarantee optimal operation 0 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
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. 0 Internal oscillator is powered up 1 Internal oscillator is powered down
2 DCDC_SEL_CLK	Select external clock for DCDC when DCDC_DISABLE_AUTO_CLK_SWITCH is set. 0 Internal oscillator is used as DCDC clock 1 External oscillator is used as DCDC clock
1 DCDC_DISABLE_AUTO_CLK_SWITCH	Disable automatic clock switch from internal oscillator to external clock. 0 Automatic clock switch feature is enabled 1 Automatic clock switch feature is disabled
0 Reserved	This field is reserved. Reserved

16.7.3 DCDC REGISTER 1 (DCDC_REG1)

Address: 4005_A000h base + 4h offset = 4005_A004h



DCDC_REG1 field descriptions

Field	Description
31–25 Reserved	This field is reserved. Reserved
24 DCDC_ LOOPCTRL_EN_ DF_HYST	Enable hysteresis in switching converter differential mode analog comparators. This feature improves transient supply ripple and efficiency. **NOTE:** Value of this bit needs to be set as explained in Application Requirements.
23 DCDC_ LOOPCTRL_EN_ CM_HYST	Enable hysteresis in switching converter common mode analog comparators. This feature improves transient supply ripple and efficiency. **NOTE:** Value of this bit needs to be set as explained in Application Requirements.
22 DCDC_ LOOPCTRL_DF_ HST_THRESH	Enable hysteresis in switching converter differential mode analog comparators. This feature improves transient supply ripple and efficiency. **NOTE:** Value of this bit needs to be set as explained in Application Requirements.
21 DCDC_ LOOPCTRL_ CM_HST_ THRESH	Enable hysteresis in switching converter common mode analog comparators. This feature improves transient supply ripple and efficiency. **NOTE:** Value of this bit needs to be set as explained in Application Requirements.
20–14 Reserved	This field is reserved. Reserved.

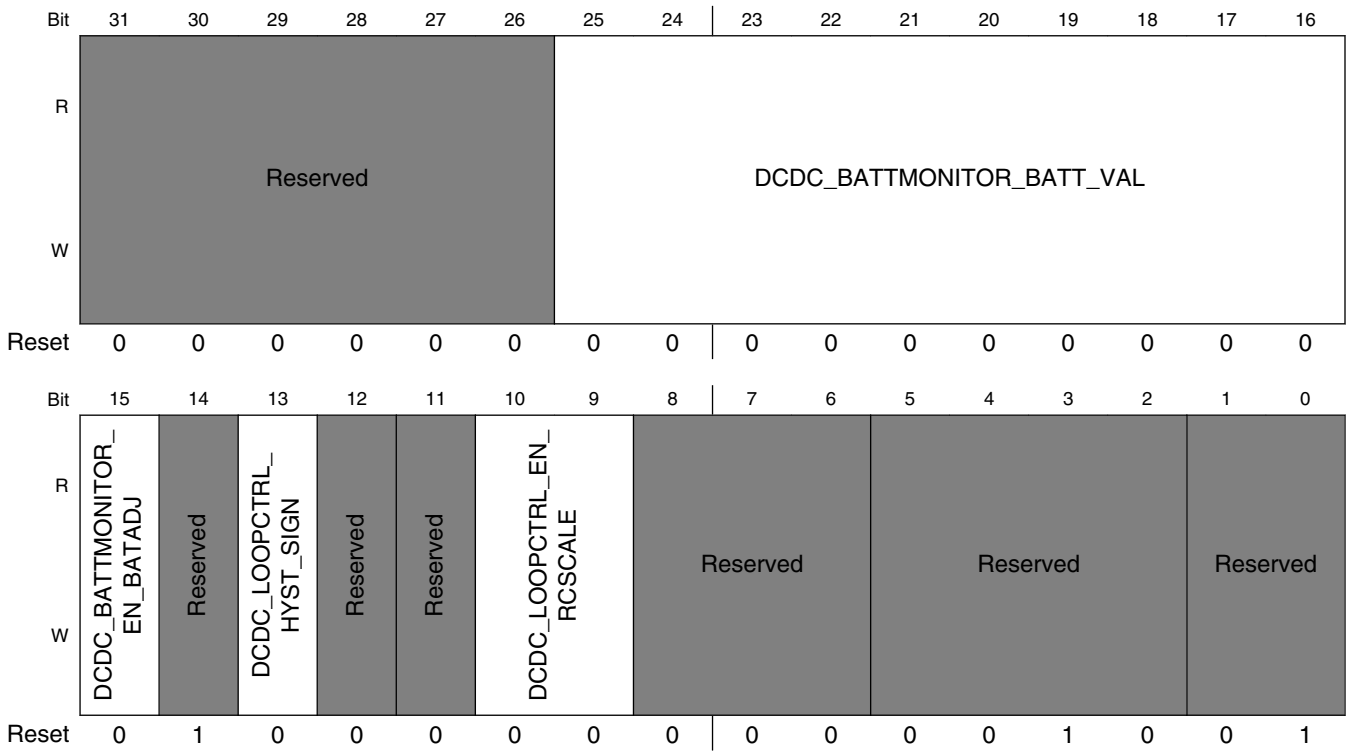
Table continues on the next page...

DCDC_REG1 field descriptions (continued)

Field	Description
13–7 Reserved	This field is reserved. Reserved
POSLIMIT_ BUCK_IN	Upper limit duty cycle limit in DCDC 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.

16.7.4 DCDC REGISTER 2 (DCDC_REG2)

Address: 4005_A000h base + 8h offset = 4005_A008h



DCDC_REG2 field descriptions

Field	Description
31–26 Reserved	This field is reserved. Reserved
25–16 DCDC_ BATTMONITOR_ BATT_VAL	Software should write the VDCDC_IN in this register measured with an 8 mV LSB resolution through the ADC. This value is used by the DCDC converter and must be properly configured before setting EN_BATADJ. The ADC should use the internal battery voltage channel to measure VDCDC_IN. Refer to the chip configuration chapter for battery voltage ADC channel assignment.
15 DCDC_ BATTMONITOR_ EN_BATADJ	This bit enables the DCDC to improve efficiency and minimize ripple using the information from the BATT_VAL field. The BATT_VAL contains accurate information before setting EN_BATADJ. The ADC should use the internal battery voltage channel to measure VDCDC_IN. Refer to the chip configuration chapter for battery voltage ADC channel assignment.

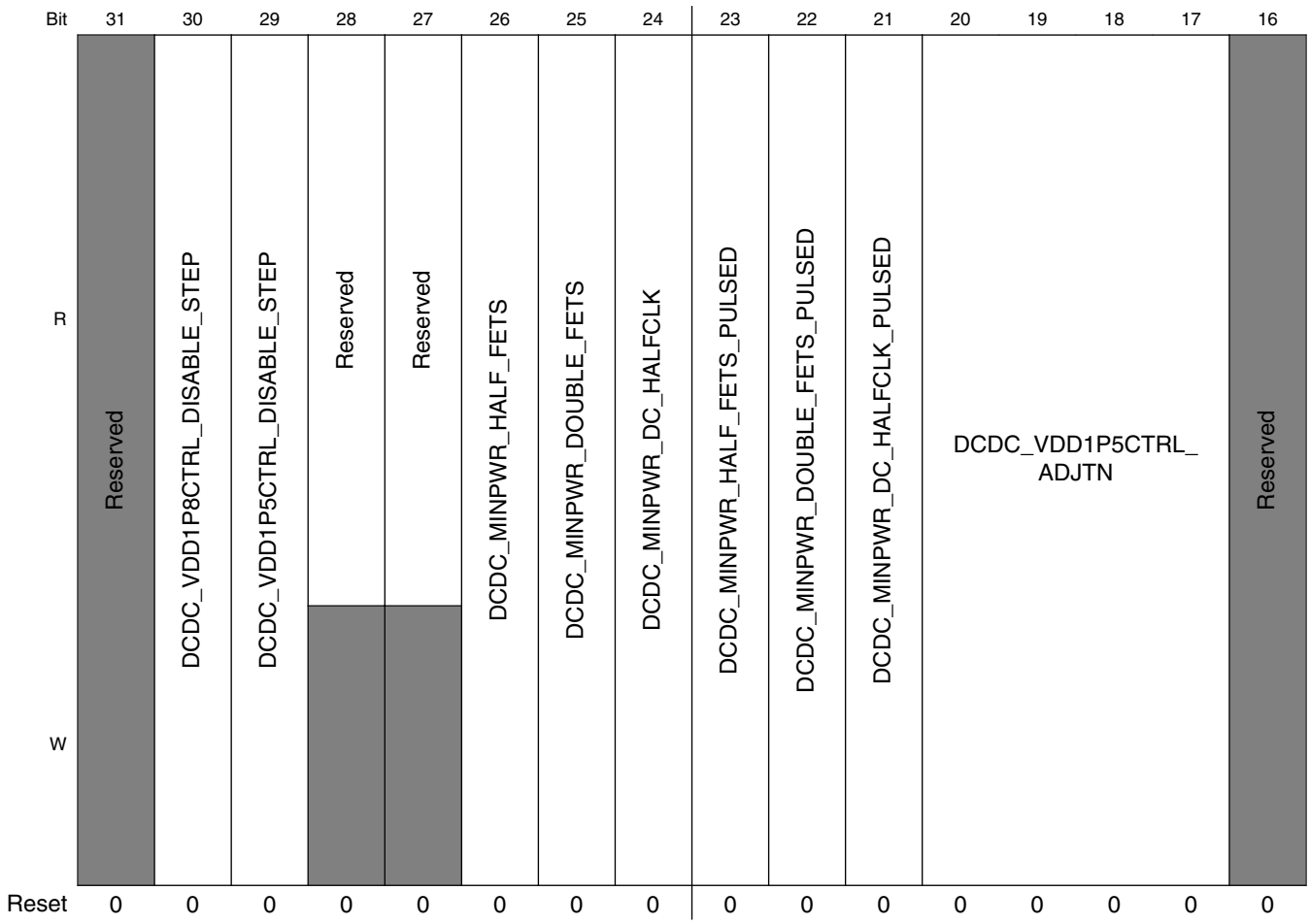
Table continues on the next page...

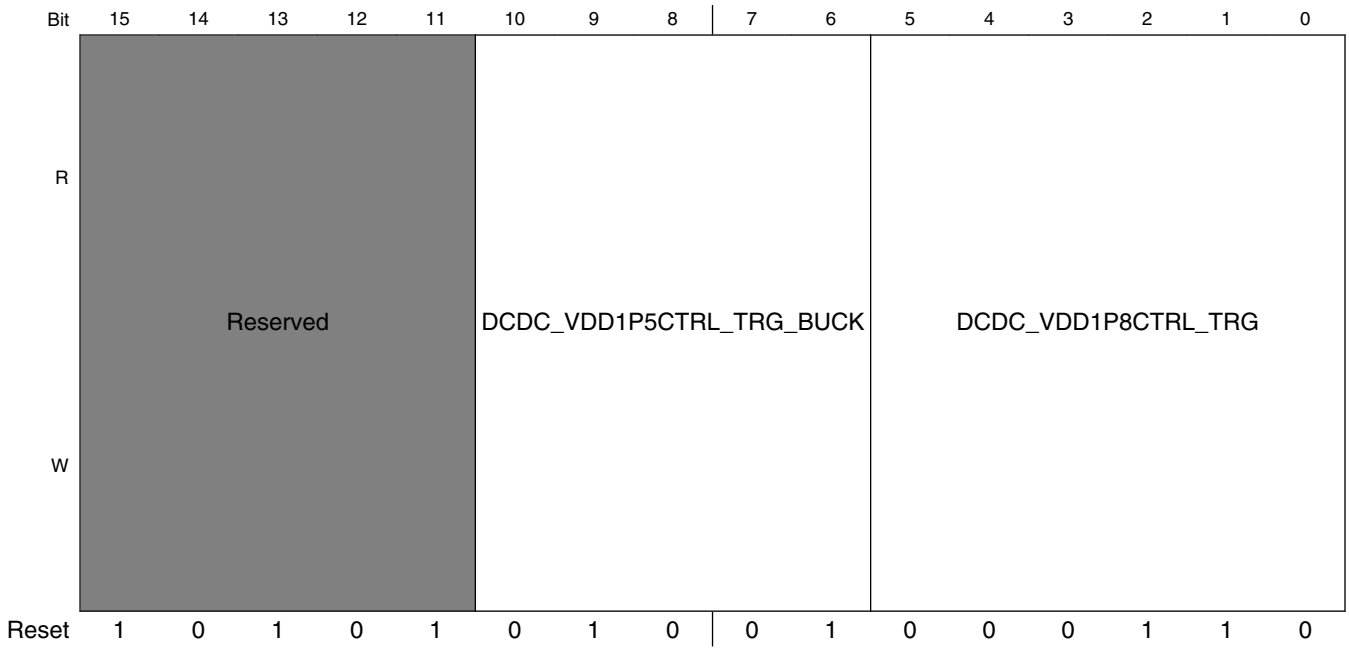
DCDC_REG2 field descriptions (continued)

Field	Description
	0 Disable the usage of the DCDC_BATTMONITOR_BATT_VAL value to calculate DCDC loop control 1 Enable the usage of DCDC_BATTMONITOR_BATT_VAL value to calculate DCDC loop control
14 Reserved	This field is reserved. Reserved
13 DCDC_ LOOPCTRL_ HYST_SIGN	This bit ensures proper switching of DCDC in Pulsed mode and is set in Pulsed mode. 0 Hysteresis sign not inverted 1 Hysteresis sign inverted (proper switching gauranteed)
12 Reserved	This field is reserved. Reserved
11 Reserved	This field is reserved. Reserved
10–9 DCDC_ LOOPCTRL_EN_ RCSCALE	The DCDC_LOOPCTRL_EN_RCSCALE reduces the response time of the DCDC to transient loads. 00 Default response time 01 2 times faster than default 10 4 times faster than default 11 8 times faster than default
8–6 Reserved	This field is reserved. Reserved
5–2 Reserved	This field is reserved. Reserved
Reserved	This field is reserved. Reserved

16.7.5 DCDC REGISTER 3 (DCDC_REG3)

Address: 4005_A000h base + Ch offset = 4005_A00Ch





DCDC_REG3 field descriptions

Field	Description
31 Reserved	This field is reserved. Reserved
30 DCDC_ VDD1P8CTRL_ DISABLE_STEP	Disable stepping for VDD_1P8. Must set this bit before enter low power modes. 0 VDD_1P8 stepping enabled 1 VDD_1P8 stepping disabled
29 DCDC_ VDD1P5CTRL_ DISABLE_STEP	Disable stepping for VDD_1P5. Must set this bit before enter low power modes. 0 VDD_1P5 stepping enabled 1 VDD_1P5 stepping disabled
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. **NOTE:** Both DCDC_MINPWR_DOUBLE_FETS and DCDC_MINPWR_HALF_FETS fields cannot be set to 1 at the same time. 0 Normal operation 1 Use Half FET
25 DCDC_ MINPWR_ DOUBLE_FETS	Use double switch FET for the continuous mode. **NOTE:** Both DCDC_MINPWR_DOUBLE_FETS and DCDC_MINPWR_HALF_FETS fields cannot be set to 1 at the same time. 0 Normal operation 1 Use Double FET

Table continues on the next page...

DCDC_REG3 field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>24
DCDC_
MINPWR_DC_
HALFCLK</td><td>Set DCDC clock to half frequency for the continuous mode.

0 Normal operation for DCDC clock
1 DCDC clock operates at half frequency</td></tr>
<tr><td>23
DCDC_
MINPWR_HALF_
FETS_PULSED</td><td>Use half switch FET for the Pulsed mode.

NOTE: Both DCDC_MINPWR_DOUBLE_FETS_PULSED and DCDC_MINPWR_HALF_FETS_PULSED fields cannot be set to 1 at the same time.

0 Pulsed mode uses normal output configuration
1 Pulsed mode uses half FET output configuration</td></tr>
<tr><td>22
DCDC_
MINPWR_
DOUBLE_FETS_
PULSED</td><td>Use double switch FET for the Pulsed mode.

NOTE: Both DCDC_MINPWR_DOUBLE_FETS_PULSED and DCDC_MINPWR_HALF_FETS_PULSED fields cannot be set to 1 at the same time.

0 Pulsed mode uses normal output configuration
1 Pulsed mode uses double FET output configuration</td></tr>
<tr><td>21
DCDC_
MINPWR_DC_
HALFCLK_
PULSED</td><td>Set DCDC clock to half frequency for the Pulsed mode.

0 Pulsed mode uses normal operation for DCDC clock
1 Pulsed mode uses half frequency DCDC clock operation</td></tr>
<tr><td>20–17
DCDC_
VDD1P5CTRL_
ADJTN</td><td>Adjust value of duty cycle when switching between VDD_1P5 and VDD_1P8. The unit is 1/32 or 3.125%.

NOTE: When using the battery monitor feature it is not necessary to set this bit. We recommend using the battery monitor feature.</td></tr>
<tr><td>16
Reserved</td><td>This field is reserved.
Reserved</td></tr>
<tr><td>15–11
Reserved</td><td>This field is reserved.
Reserved</td></tr>
<tr><td>10–6
DCDC_
VDD1P5CTRL_
TRG_BUCK</td><td>Target value of VDD_1P5 in buck mode, 25 mV each step from 0x00 to 0x0F

<table>
<tr><th>Code</th><th>VDD_1P8 Output Target (V)</th></tr>
<tr><td>0x00</td><td>1.275 V</td></tr>
<tr><td>0x01</td><td>1.300 V</td></tr>
<tr><td>0x02</td><td>1.325 V</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>0x0F</td><td>1.65 V</td></tr>
</table>

NOTE: Before writing this register, these conditions must be met:
1. DCDC must be in Continuous Mode
2. DCDC_STS_DC_OK bit must be set</td></tr>
<tr><td>DCDC_
VDD1P8CTRL_
TRG</td><td>Target value of VDD_1P8 :

25 mV each step in two ranges, from 0x00 to 0x11 and 0x20 to 0x3F and 50 mV each step in range from 0x12 to 0x1F.</td></tr>
</table>

Table continues on the next page...

DCDC_REG3 field descriptions (continued)

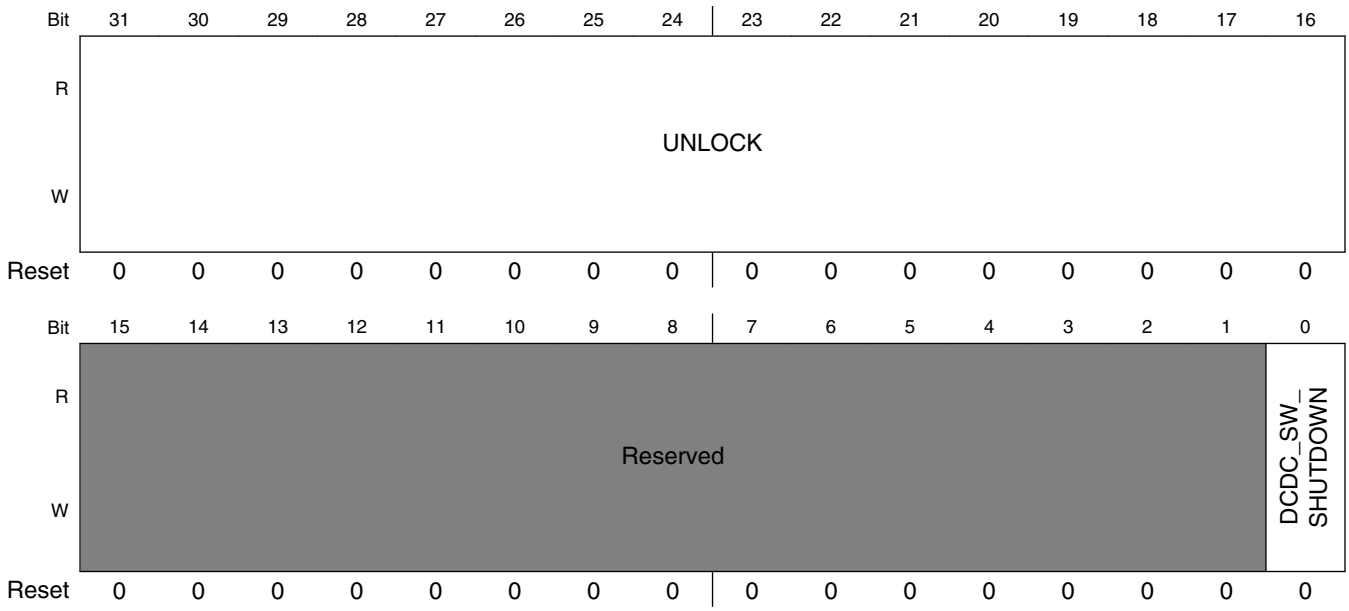
Field	Description

Code	VDD_1P8 Output Target (V)
0x00	1.650 V
0x01	1.675 V
0x02	1.700 V
...	...
0x11	2.075 V
0x12	2.100 V
0x13	2.150 V
0x14	2.200 V
0x15	2.250 V
...	...
0x1F	2.750 V
0x20	2.800 V
0x21	2.825 V
0x22	2.850 V
...	...
0x3C	3.500 V

NOTE: Before writing this register, these conditions must be met:
1. DCDC must be in Continuous Mode
2. DCDC_STS_DC_OK bit must be set

16.7.6 DCDC REGISTER 4 (DCDC_REG4)

Address: 4005_A000h base + 10h offset = 4005_A010h

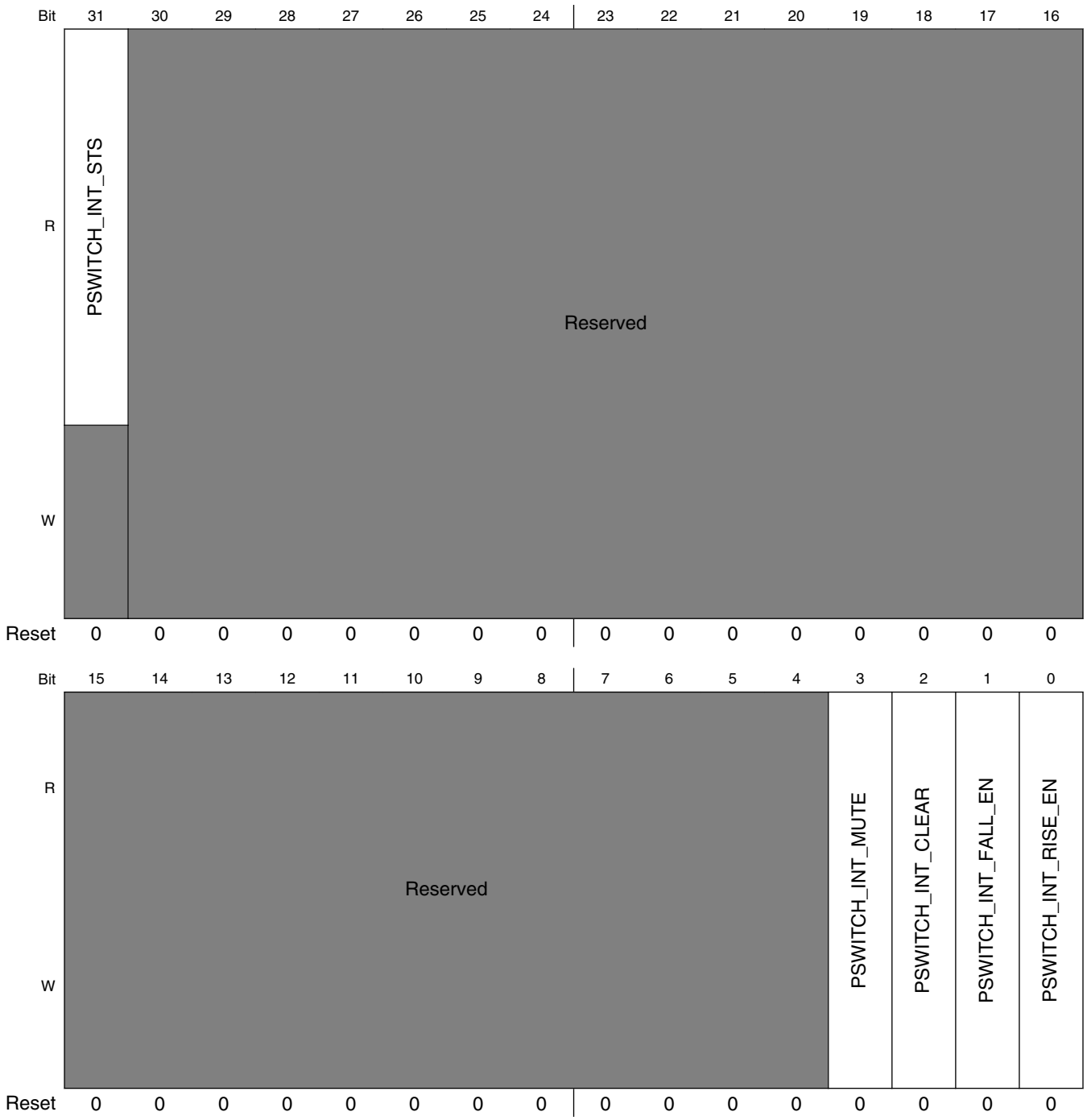


DCDC_REG4 field descriptions

Field	Description
31–16 UNLOCK	0x3E77 KEY—Key needed to unlock DCDC_REG4 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 field must 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 (DCDC Turn on time (T_{DCDC_ON}; refer to the data sheet for specific time). **NOTE** This bit should not be used in buck mode when PSWITCH is tied to DCDC_IN.

16.7.7 DCDC REGISTER 6 (DCDC_REG6)

Address: 4005_A000h base + 18h offset = 4005_A018h

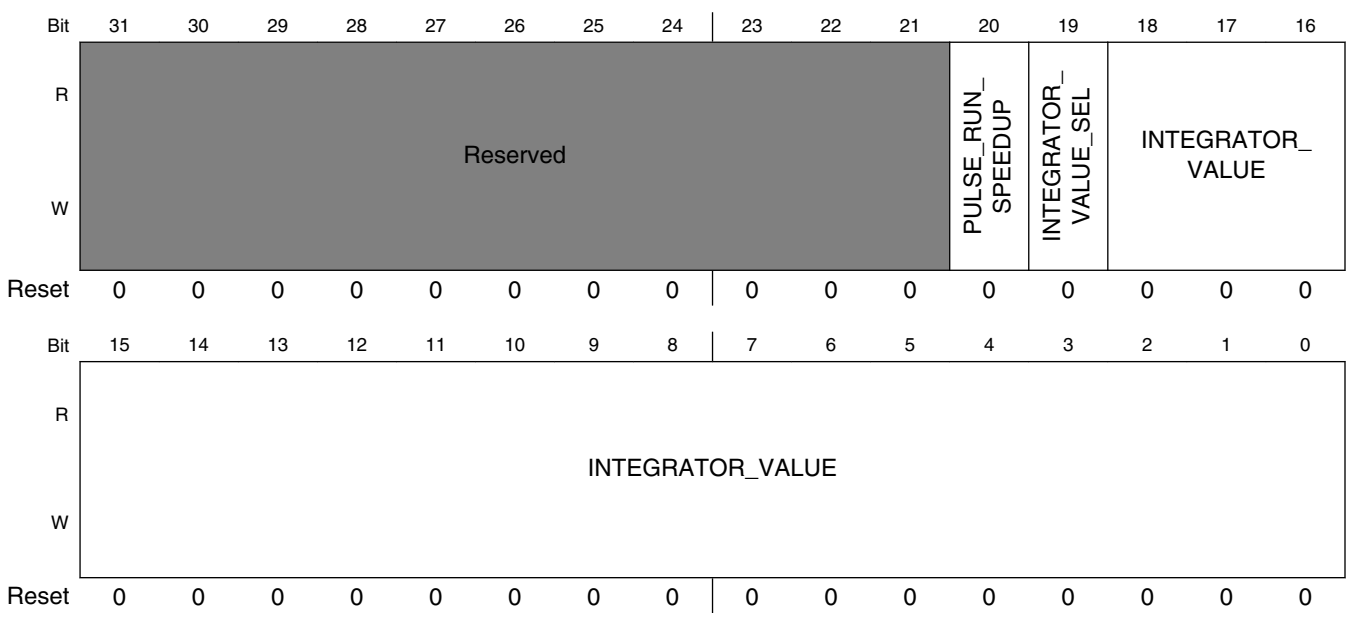


DCDC_REG6 field descriptions

Field	Description
31 PSWITCH_INT_STS	PSWITCH edge detection interrupt status. This bit indicates whether or not an interrupt has occurred. This bit is cleared by writing to the PSWITCH_INT_CLEAR bit (DCDC_REG6[PSWITCH_INT_CLEAR]) 0 PSWITCH interrupt has not occurred 1 PSWITCH interrupt has occurred
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	This bit clears the PSWITCH interrupt. 0 No effect 1 Clear PSWITCH interrupt
1 PSWITCH_INT_FALL_EN	Enable falling edge detect for interrupt. 0 PSWITCH falling edge interrupt disabled 1 PSWITCH falling edge interrupt enabled
0 PSWITCH_INT_RISE_EN	Enable rising edge detect for interrupt. 0 PSWITCH rising edge interrupt disabled 1 PSWITCH rising edge interrupt enabled

16.7.8 DCDC REGISTER 7 (DCDC_REG7)

Address: 4005_A000h base + 1Ch offset = 4005_A01Ch



DCDC_REG7 field descriptions

Field	Description
31–21 Reserved	This field is reserved. Reserved
20 PULSE_RUN_SPEEDUP	Enable pulse run speedup. Before setting this bit, INTEGRATOR_VALUE_SEL must be set to 1 and integrator value must be programmed. 0 Pulse run speedup feature disabled 1 Pulse run speedup feature enabled
19 INTEGRATOR_VALUE_SEL	Select the integrator value from above register or saved value in hardware. 0 Select the saved value in hardware. 1 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 VDD_1P5 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 = VDD_1P5 / DCDC_IN.

Chapter 17
Low-Leakage Wakeup Unit (LLWU)

17.1 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.

17.1.1 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.

17.1.2 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].

17.1.2.1 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.

17.1.2.2 VLLS modes

All wakeup and reset events result in VLLS exit via a reset flow.

17.1.2.3 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.

17.1.2.4 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.

17.1.3 Block diagram

The following figure is the block diagram for the LLWU module.

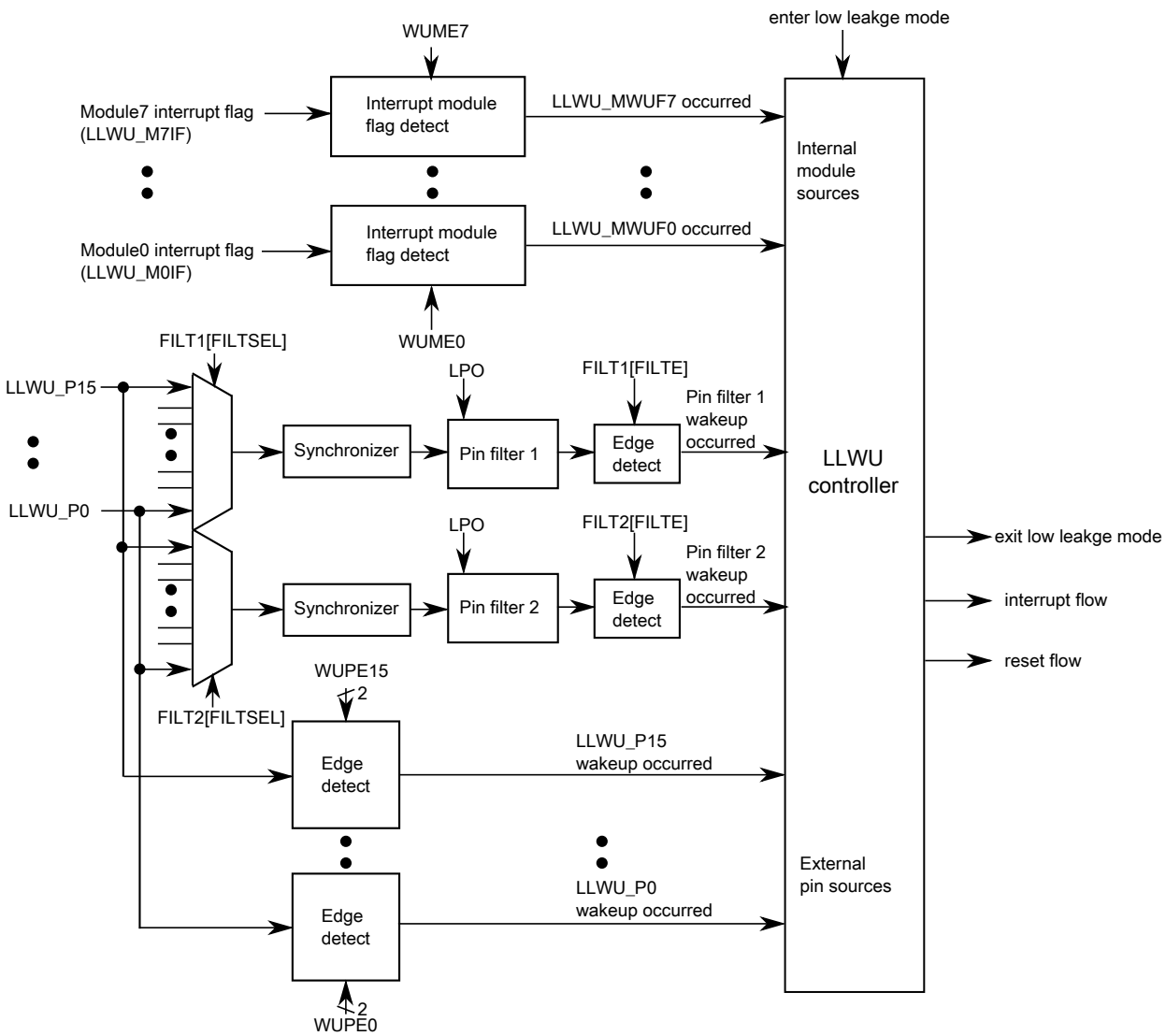


Figure 17-1. LLWU block diagram

17.2 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 17-1. LLWU signal descriptions

Signal	Description	I/O
LLWU_Pn	Wakeup inputs (n = 0-15)	I

17.3 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	17.3.1/393
4007_C001	LLWU Pin Enable 2 register (LLWU_PE2)	8	R/W	00h	17.3.2/394
4007_C002	LLWU Pin Enable 3 register (LLWU_PE3)	8	R/W	00h	17.3.3/395
4007_C003	LLWU Pin Enable 4 register (LLWU_PE4)	8	R/W	00h	17.3.4/396
4007_C004	LLWU Module Enable register (LLWU_ME)	8	R/W	00h	17.3.5/397
4007_C005	LLWU Flag 1 register (LLWU_F1)	8	R/W	00h	17.3.6/399
4007_C006	LLWU Flag 2 register (LLWU_F2)	8	R/W	00h	17.3.7/401
4007_C007	LLWU Flag 3 register (LLWU_F3)	8	R	00h	17.3.8/402
4007_C008	LLWU Pin Filter 1 register (LLWU_FILT1)	8	R/W	00h	17.3.9/404
4007_C009	LLWU Pin Filter 2 register (LLWU_FILT2)	8	R/W	00h	17.3.10/405

17.3.1 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

17.3.2 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

17.3.3 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

17.3.4 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

17.3.5 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

17.3.6 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

17.3.7 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

17.3.8 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

17.3.9 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

17.3.10 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

17.4 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.

17.4.1 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.

17.4.2 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.

17.4.3 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 18
Reset Control Module (RCM)

18.1 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.

18.2 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	18.2.1/410
4007_F001	System Reset Status Register 1 (RCM_SRS1)	8	R	00h	18.2.2/411
4007_F004	Reset Pin Filter Control register (RCM_RPFC)	8	R/W	00h	18.2.3/412
4007_F005	Reset Pin Filter Width register (RCM_RPFW)	8	R/W	00h	18.2.4/413

18.2.1 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

18.2.2 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.

18.2.3 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

18.2.4 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 19
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.

19.1 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 0x4100_0000. 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.

19.1.1 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

19.1.2 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.

19.2 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 0x4100_0000 plus a 4 KB space based at 0x4100_F000 for GPIO accesses; the decorated address space is mapped to the 448 MB region located at 0x4400_0000–0x5FFF_FFFF.

19.3 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.

19.3.1 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:

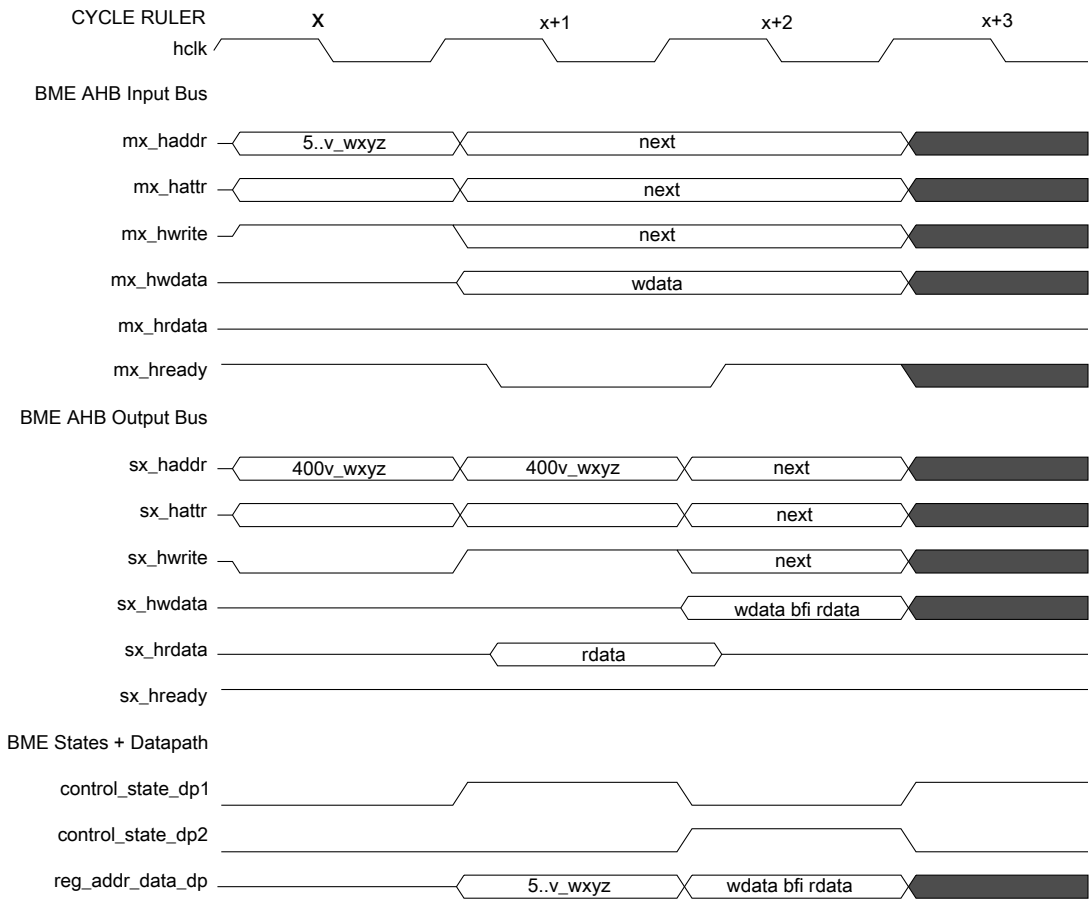


Figure 19-1. Decorated store: bit field insert timing diagram

All the decorated store operations follow the same execution template shown in Figure 19-1, 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.

19.3.1.1 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.

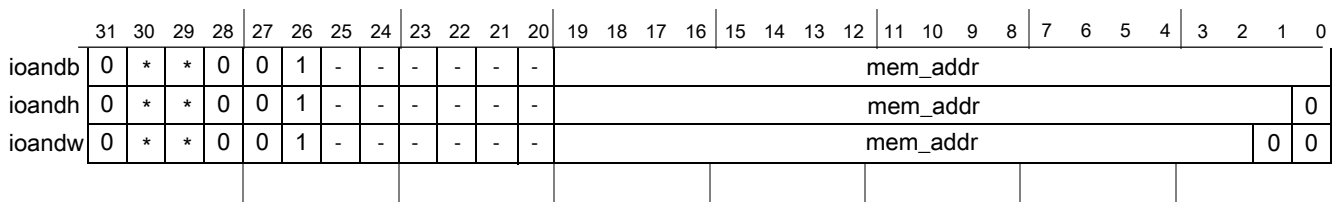


Figure 19-2. Decorated store address: logical AND

See Figure 19-2, 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 19-1. 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 19-1. 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

19.3.1.2 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.

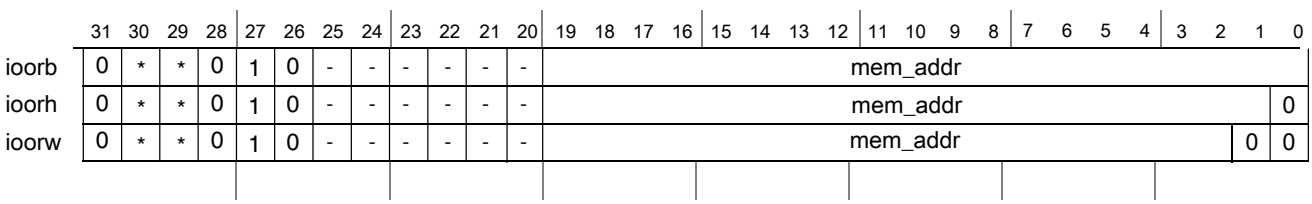


Figure 19-3. Decorated address store: logical OR

See Figure 19-3,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 19-2. 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

19.3.1.3 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.

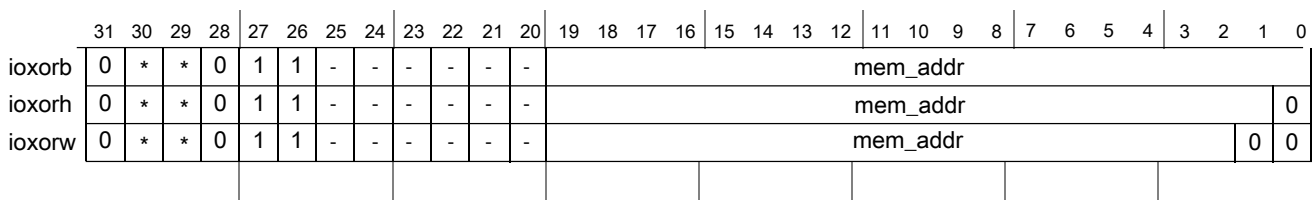


Figure 19-4. Decorated address store: logical XOR

See Figure 19-4, 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 19-3. 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

19.3.1.4 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.

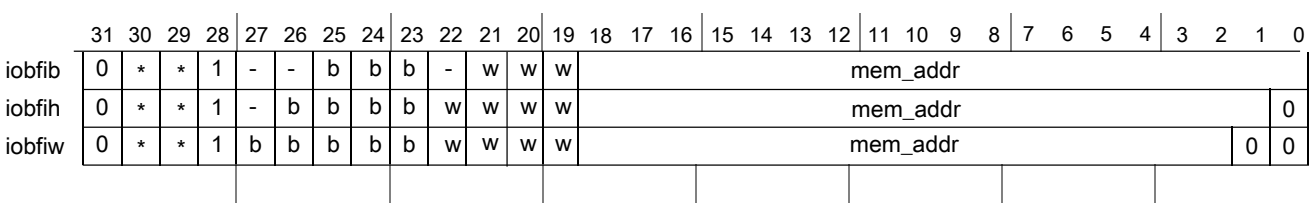


Figure 19-5. Decorated address store: bit field insert

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 19-4. 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

19.3.2 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.

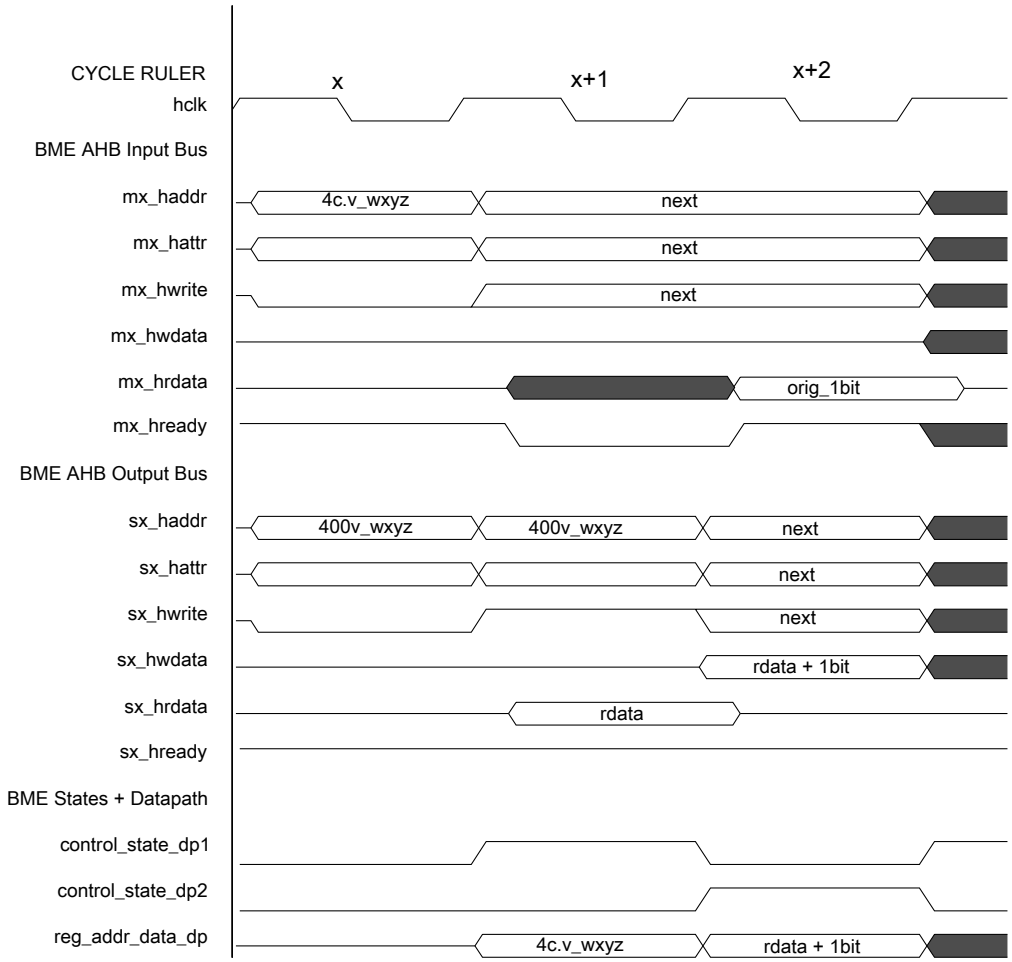


Figure 19-6. Decorated load: load-and-set 1-bit field insert timing diagram

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.

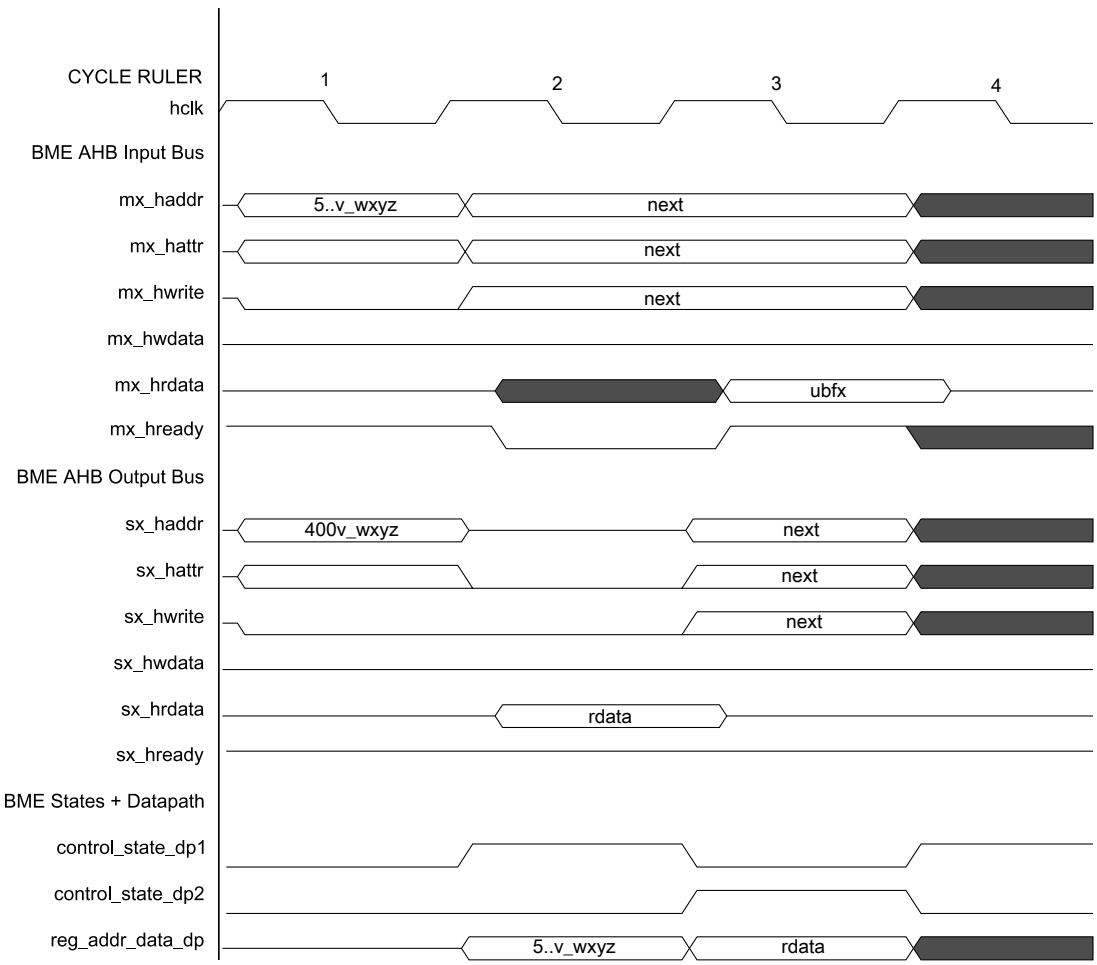


Figure 19-7. Decorated load: unsigned bit field insert timing diagram

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.

19.3.2.1 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).

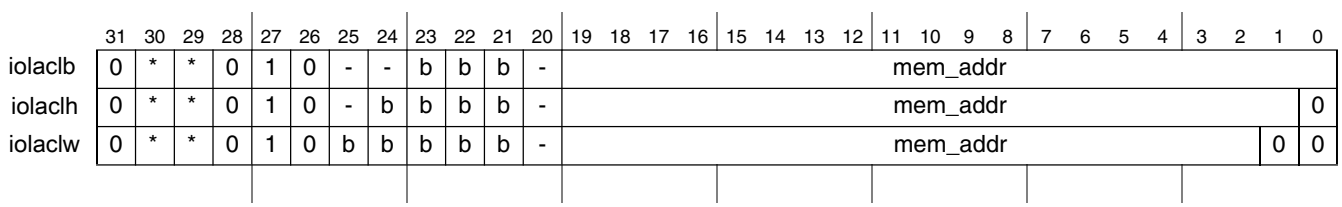


Figure 19-8. Decorated load address: load-and-clear 1 bit

See Figure 19-8, 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 19-5. 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

19.3.2.2 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).

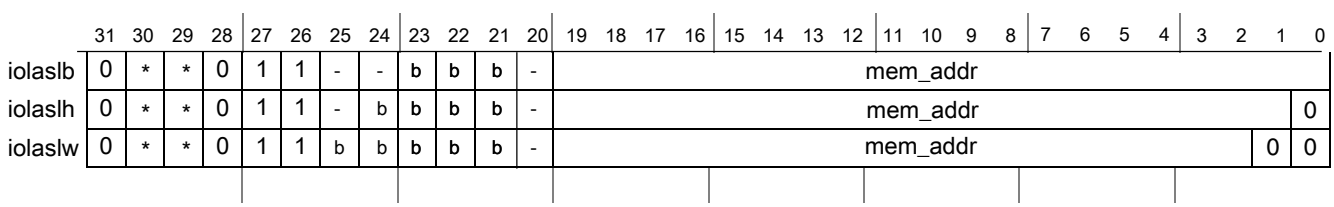


Figure 19-9. Decorated load address: load-and-set 1 bit

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 19-6. 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 | mask) and capture destination data in register	Return extracted bit to master; Perform write sending registered data to memory

19.3.2.3 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.

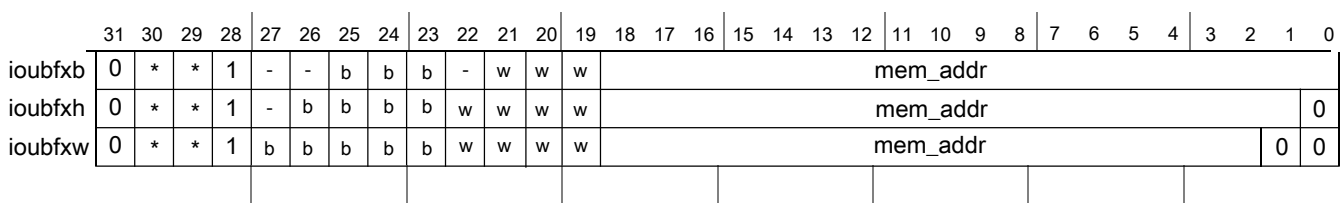


Figure 19-10. Decorated load address: unsigned bit field extract

See Figure 19-10, 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 19-7. 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

19.3.3 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 19-8. 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

19.4 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 20
Miscellaneous Control Module (MCM)

The Miscellaneous Control Module (MCM) provides a myriad of miscellaneous control functions for the platform (RPP).

20.1 Introduction

The Miscellaneous Control Module (MCM) provides a myriad of miscellaneous control functions.

20.1.1 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

20.2 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	20.2.1/434

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	20.2.2/435
F000_300C	Platform Control Register (MCM_PLACR)	32	R/W	0000_0050h	20.2.3/435
F000_3040	Compute Operation Control Register (MCM_CPO)	32	R/W	0000_0000h	20.2.4/438

20.2.1 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

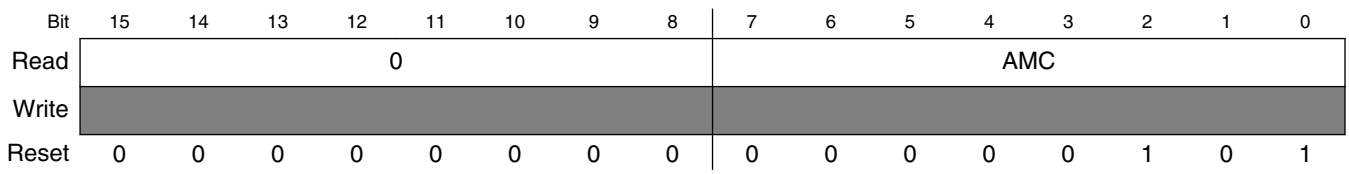
MCM_PLASC field descriptions

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.

20.2.2 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



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

20.2.3 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].

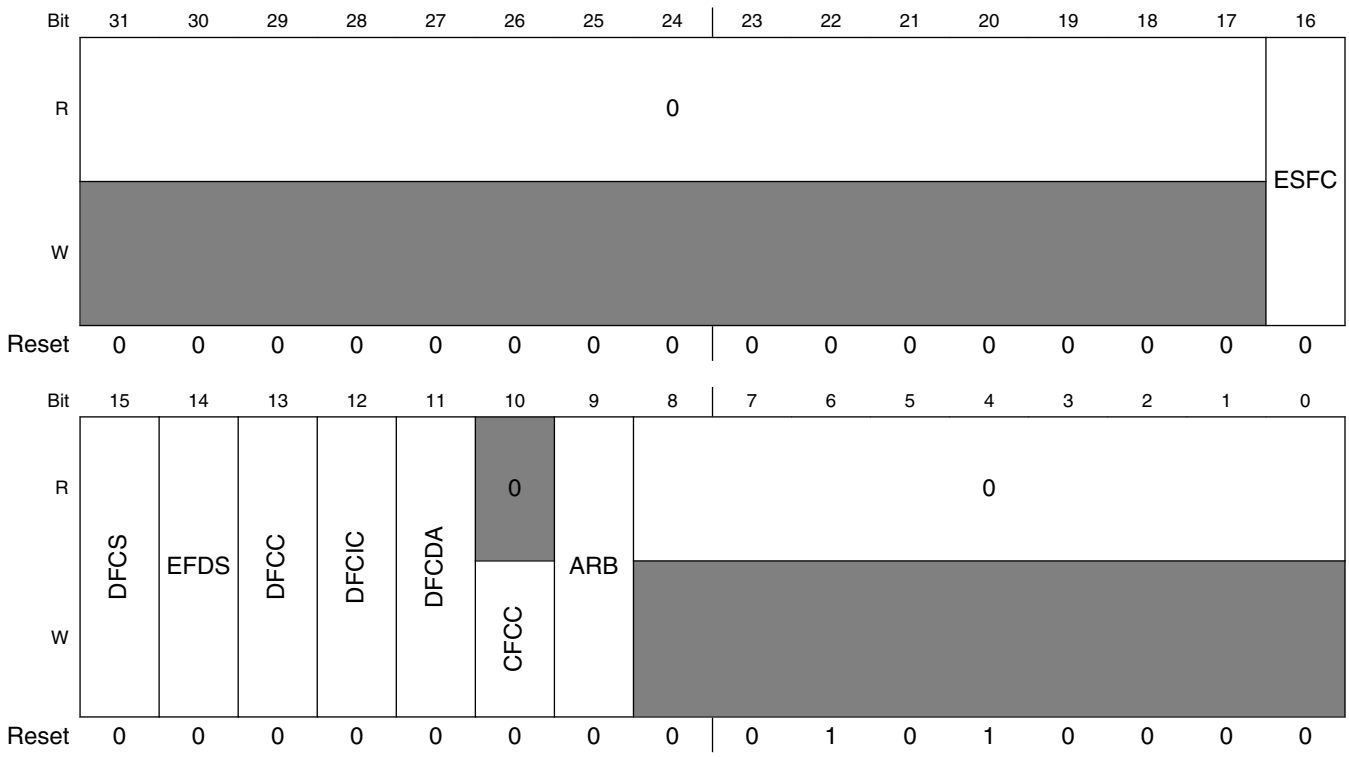
The speculation buffer is enabled only for instructions after reset. It is possible to have these states for the speculation buffer:

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



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...

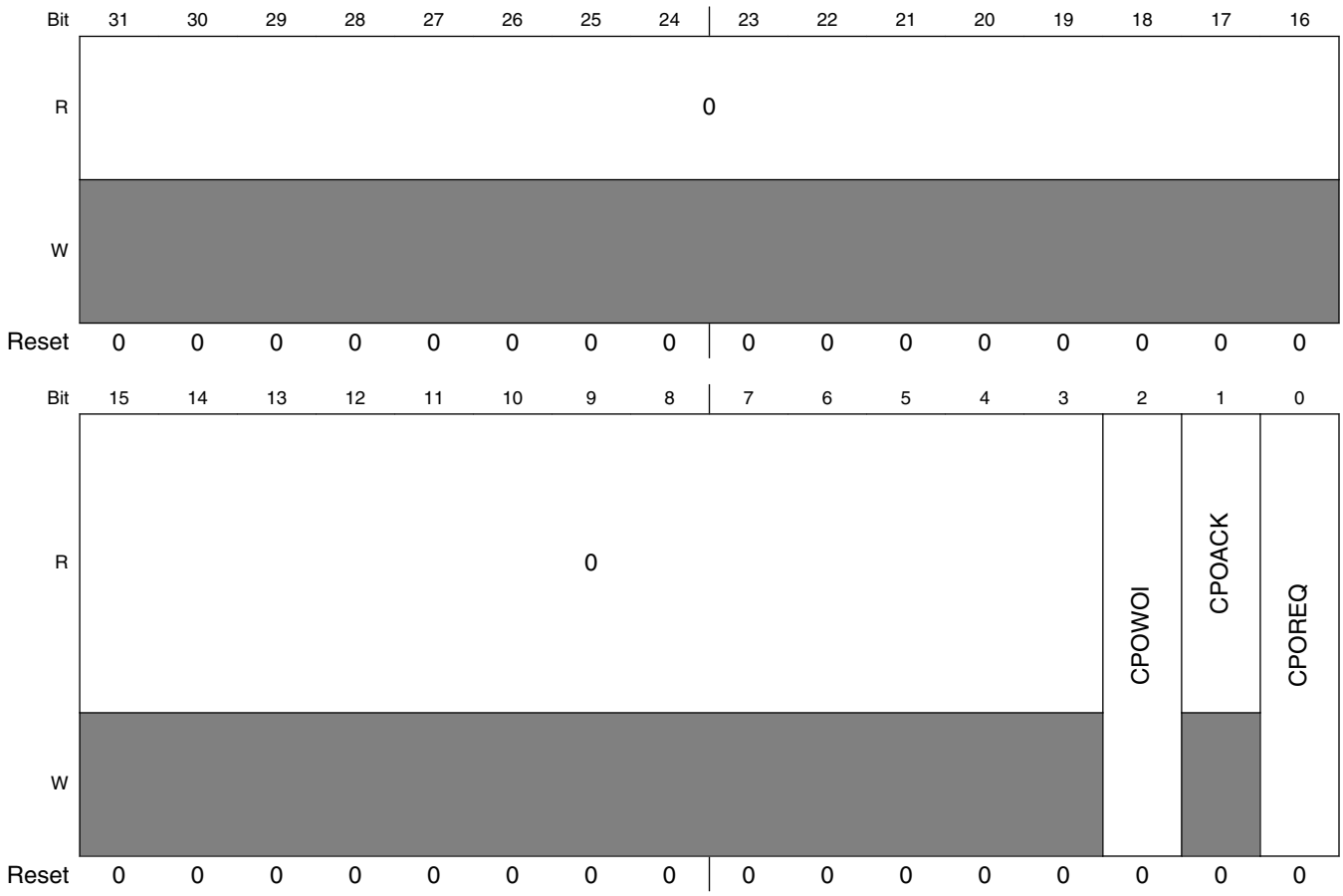
MCM_PLACR field descriptions (continued)

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.

20.2.4 Compute Operation Control Register (MCM_CPO)

This register controls the Compute Operation.

Address: F000_3000h base + 40h offset = F000_3040h



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 21
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.

21.1 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.

21.1.1 Overview

A generic block diagram of the processor core and platform for this class of ultra low-end microcontrollers is shown as follows:

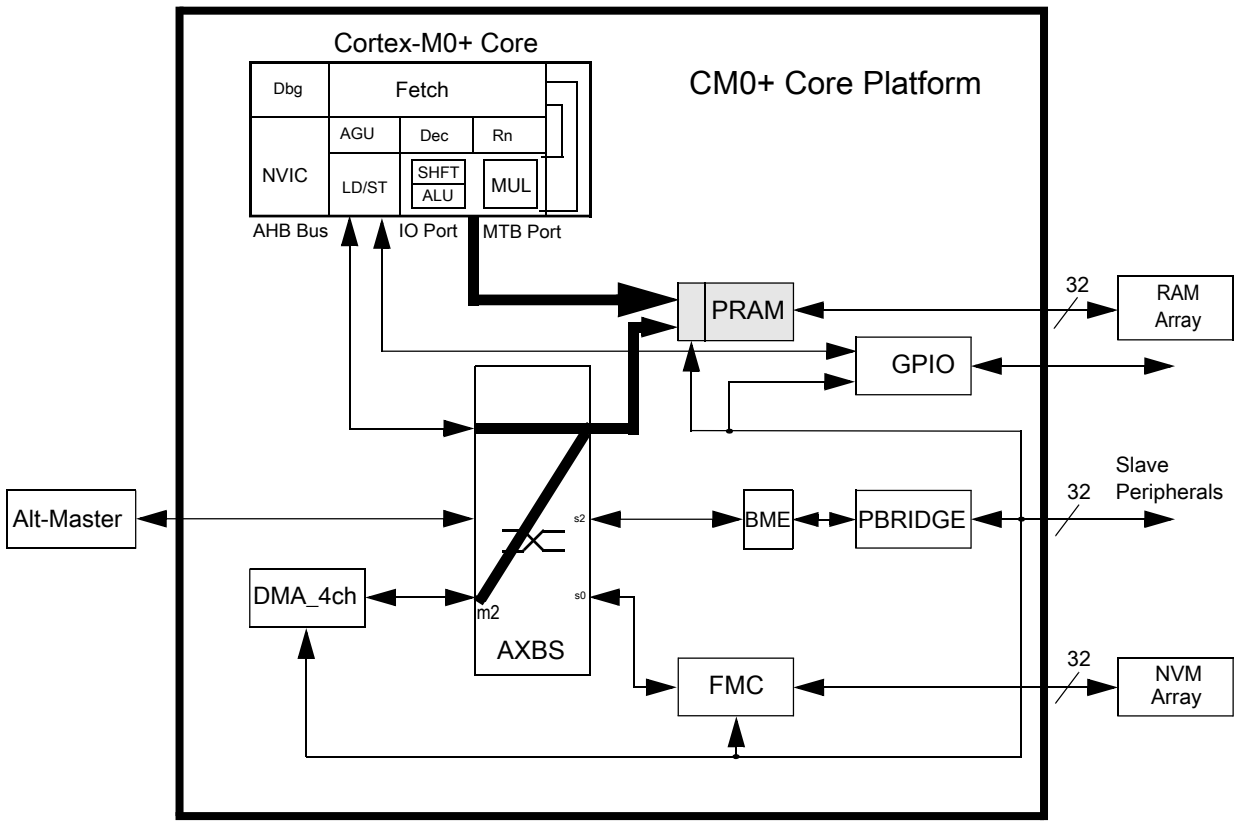


Figure 21-1. Cortex-M0+ core platform block diagram

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.

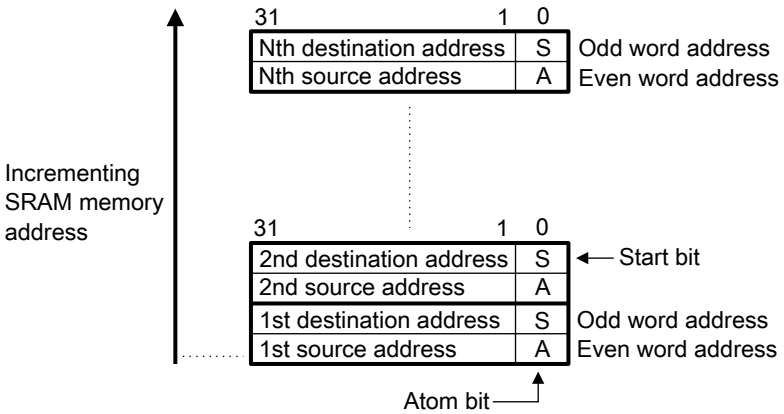


Figure 21-2. MTB execution trace storage format

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.

21.1.2 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

21.1.3 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.

21.2 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 21-1. 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 21-1. 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.

21.3 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

21.3.1 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	21.3.1.1/ 448
F000_0004	MTB Master Register (MTB_MASTER)	32	R/W	See section	21.3.1.2/ 450
F000_0008	MTB Flow Register (MTB_FLOW)	32	R/W	Undefined	21.3.1.3/ 451
F000_000C	MTB Base Register (MTB_BASE)	32	R	Undefined	21.3.1.4/ 453
F000_0F00	Integration Mode Control Register (MTB_MODECTRL)	32	R	0000_0000h	21.3.1.5/ 454
F000_0FA0	Claim TAG Set Register (MTB_TAGSET)	32	R	0000_0000h	21.3.1.6/ 454
F000_0FA4	Claim TAG Clear Register (MTB_TAGCLEAR)	32	R	0000_0000h	21.3.1.7/ 455
F000_0FB0	Lock Access Register (MTB_LOCKACCESS)	32	R	0000_0000h	21.3.1.8/ 455
F000_0FB4	Lock Status Register (MTB_LOCKSTAT)	32	R	0000_0000h	21.3.1.9/ 456
F000_0FB8	Authentication Status Register (MTB_AUTHSTAT)	32	R	0000_0000h	21.3.1.10/ 456
F000_0FBC	Device Architecture Register (MTB_DEVICEARCH)	32	R	4770_0A31h	21.3.1.11/ 457
F000_0FC8	Device Configuration Register (MTB_DEVICECFG)	32	R	0000_0000h	21.3.1.12/ 457
F000_0FCC	Device Type Identifier Register (MTB_DEVICETYPID)	32	R	0000_0031h	21.3.1.13/ 458
F000_0FD0	Peripheral ID Register (MTB_PERIPHID4)	32	R	See section	21.3.1.14/ 458
F000_0FD4	Peripheral ID Register (MTB_PERIPHID5)	32	R	See section	21.3.1.14/ 458
F000_0FD8	Peripheral ID Register (MTB_PERIPHID6)	32	R	See section	21.3.1.14/ 458
F000_0FDC	Peripheral ID Register (MTB_PERIPHID7)	32	R	See section	21.3.1.14/ 458
F000_0FE0	Peripheral ID Register (MTB_PERIPHID0)	32	R	See section	21.3.1.14/ 458
F000_0FE4	Peripheral ID Register (MTB_PERIPHID1)	32	R	See section	21.3.1.14/ 458
F000_0FE8	Peripheral ID Register (MTB_PERIPHID2)	32	R	See section	21.3.1.14/ 458
F000_0FEC	Peripheral ID Register (MTB_PERIPHID3)	32	R	See section	21.3.1.14/ 458

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	21.3.1.15/ 459
F000_0FF4	Component ID Register (MTB_COMPID1)	32	R	See section	21.3.1.15/ 459
F000_0FF8	Component ID Register (MTB_COMPID2)	32	R	See section	21.3.1.15/ 459
F000_0FFC	Component ID Register (MTB_COMPID3)	32	R	See section	21.3.1.15/ 459

21.3.1.1 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:

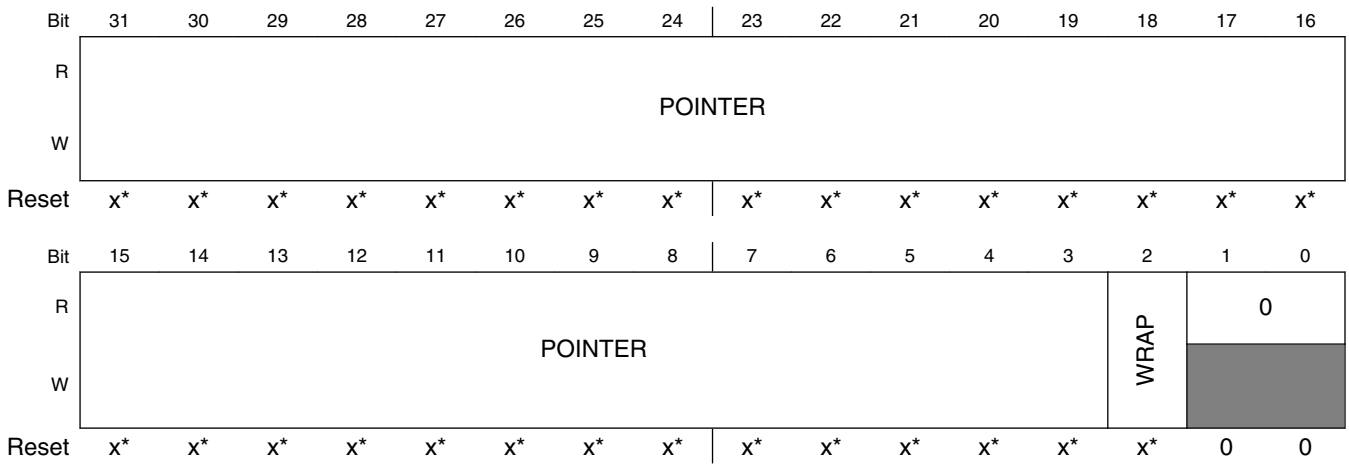
For the standard configuration where the size of the MTB is ≤ 25% of the total RAM capacity, it is recommended the MTB be based at the address defined by the MTB_BASE register. The read-only MTB_BASE register is defined by the expression (0x2000_0000 - (RAM_Size/4)). For this configuration, the MTB_POSITION 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



* 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.

21.3.1.2 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.

21.3.1.3 MTB Flow Register (MTB_FLOW)

The MTB_FLOW register contains the watermark address and the autostop/autohalt control bits.

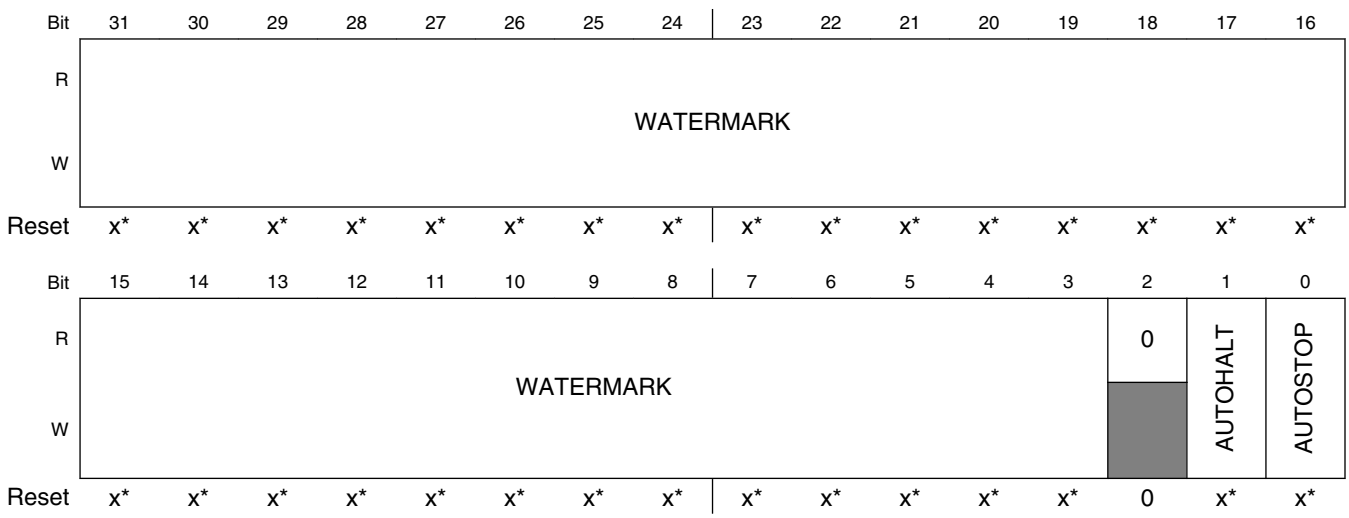
If tracing is stopped using the watermark autostop feature, it cannot be restarted until software clears the watermark autostop. This can be achieved in one of the following ways:

- 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



* 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.

21.3.1.4 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.

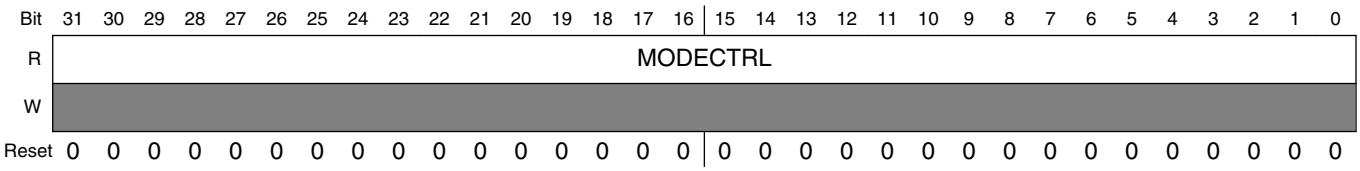
MTB_BASE field descriptions

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.

21.3.1.5 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



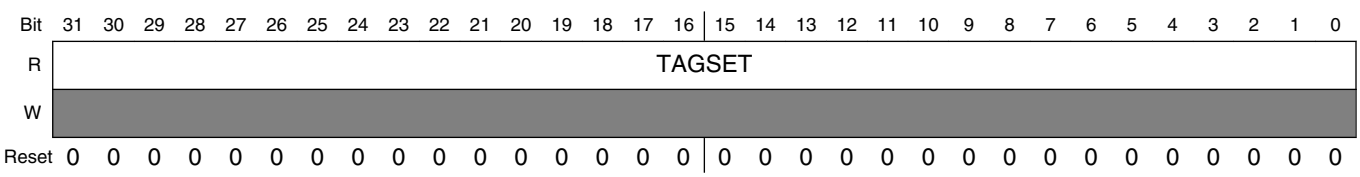
MTB_MODECTRL field descriptions

Field	Description
MODECTRL	MODECTRL Hardwired to 0x0000_0000

21.3.1.6 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



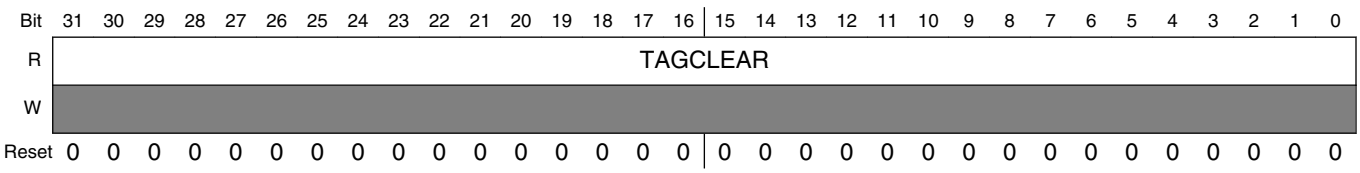
MTB_TAGSET field descriptions

Field	Description
TAGSET	TAGSET Hardwired to 0x0000_0000

21.3.1.7 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



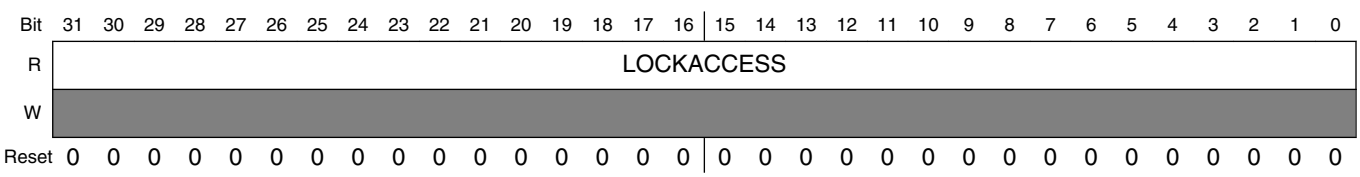
MTB_TAGCLEAR field descriptions

Field	Description
TAGCLEAR	TAGCLEAR Hardwired to 0x0000_0000

21.3.1.8 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



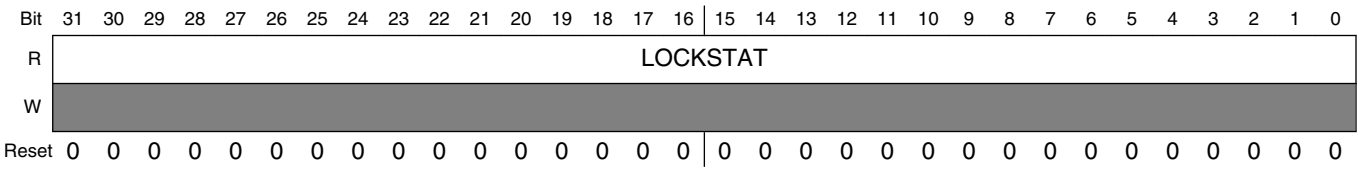
MTB_LOCKACCESS field descriptions

Field	Description
LOCKACCESS	Hardwired to 0x0000_0000

21.3.1.9 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



MTB_LOCKSTAT field descriptions

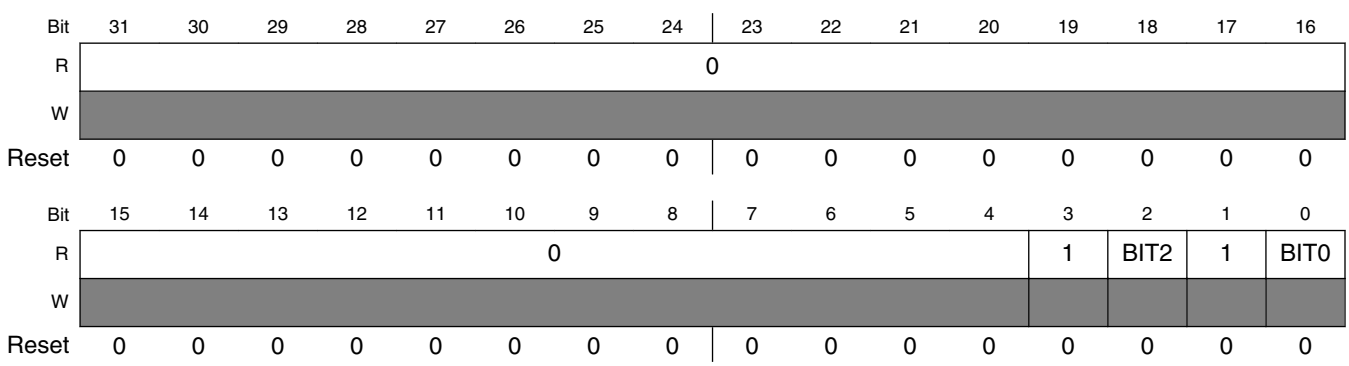
Field	Description
LOCKSTAT	LOCKSTAT Hardwired to 0x0000_0000

21.3.1.10 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



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.

21.3.1.11 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.

21.3.1.12 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.

21.3.1.13 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.

21.3.1.14 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*

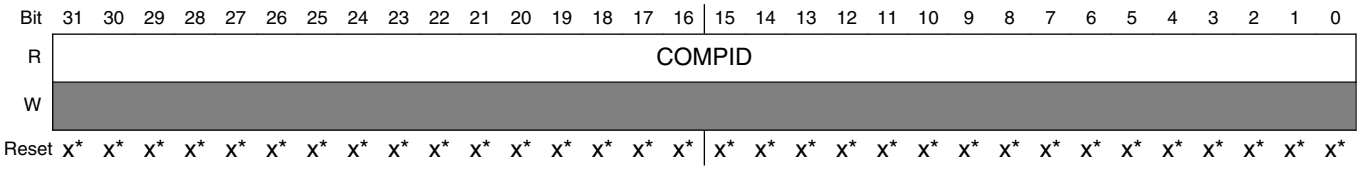
MTB_PERIPHID*n* field descriptions

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.

21.3.1.15 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



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.

21.3.2 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.

MTBDWT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_1000	MTB DWT Control Register (MTBDWT_CTRL)	32	R	2F00_0000h	21.3.2.1/ 460
F000_1020	MTB_DWT Comparator Register (MTBDWT_COMP0)	32	R/W	0000_0000h	21.3.2.2/ 461
F000_1024	MTB_DWT Comparator Mask Register (MTBDWT_MASK0)	32	R/W	0000_0000h	21.3.2.3/ 462
F000_1028	MTB_DWT Comparator Function Register 0 (MTBDWT_FCT0)	32	R/W	0000_0000h	21.3.2.4/ 463
F000_1030	MTB_DWT Comparator Register (MTBDWT_COMP1)	32	R/W	0000_0000h	21.3.2.2/ 461
F000_1034	MTB_DWT Comparator Mask Register (MTBDWT_MASK1)	32	R/W	0000_0000h	21.3.2.3/ 462
F000_1038	MTB_DWT Comparator Function Register 1 (MTBDWT_FCT1)	32	R/W	0000_0000h	21.3.2.5/ 465
F000_1200	MTB_DWT Trace Buffer Control Register (MTBDWT_TBCTRL)	32	R/W	2000_0000h	21.3.2.6/ 466

Table continues on the next page...

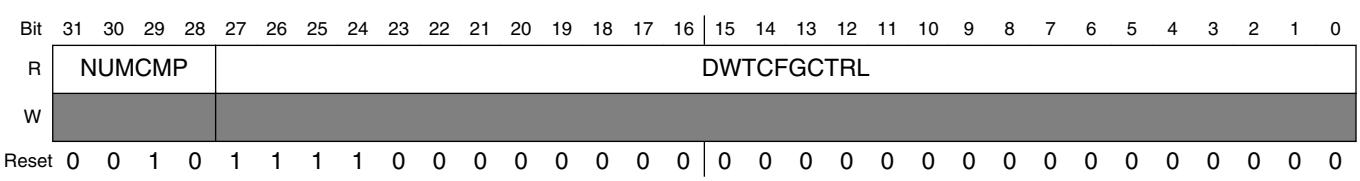
MTBDWT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_1FC8	Device Configuration Register (MTBDWT_DEVICECFG)	32	R	0000_0000h	21.3.2.7/ 468
F000_1FCC	Device Type Identifier Register (MTBDWT_DEVICETYPID)	32	R	0000_0004h	21.3.2.8/ 468
F000_1FD0	Peripheral ID Register (MTBDWT_PERIPHID4)	32	R	See section	21.3.2.9/ 469
F000_1FD4	Peripheral ID Register (MTBDWT_PERIPHID5)	32	R	See section	21.3.2.9/ 469
F000_1FD8	Peripheral ID Register (MTBDWT_PERIPHID6)	32	R	See section	21.3.2.9/ 469
F000_1FDC	Peripheral ID Register (MTBDWT_PERIPHID7)	32	R	See section	21.3.2.9/ 469
F000_1FE0	Peripheral ID Register (MTBDWT_PERIPHID0)	32	R	See section	21.3.2.9/ 469
F000_1FE4	Peripheral ID Register (MTBDWT_PERIPHID1)	32	R	See section	21.3.2.9/ 469
F000_1FE8	Peripheral ID Register (MTBDWT_PERIPHID2)	32	R	See section	21.3.2.9/ 469
F000_1FEC	Peripheral ID Register (MTBDWT_PERIPHID3)	32	R	See section	21.3.2.9/ 469
F000_1FF0	Component ID Register (MTBDWT_COMPID0)	32	R	See section	21.3.2.10/ 469
F000_1FF4	Component ID Register (MTBDWT_COMPID1)	32	R	See section	21.3.2.10/ 469
F000_1FF8	Component ID Register (MTBDWT_COMPID2)	32	R	See section	21.3.2.10/ 469
F000_1FFC	Component ID Register (MTBDWT_COMPID3)	32	R	See section	21.3.2.10/ 469

21.3.2.1 MTB DWT Control Register (MTBDWT_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



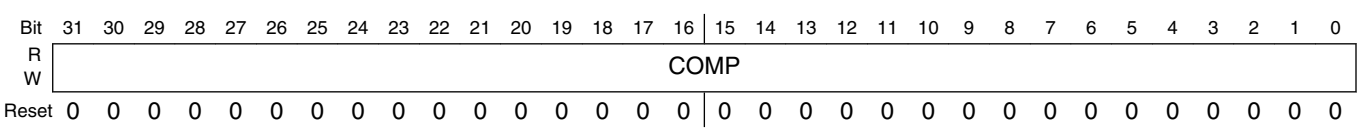
MTBDWT_CTRL field descriptions

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

21.3.2.2 MTB_DWT Comparator Register (MTBDWT_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



MTBDWT_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

MTBDWT_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".

21.3.2.3 MTB_DWT Comparator Mask Register (MTBDWT_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

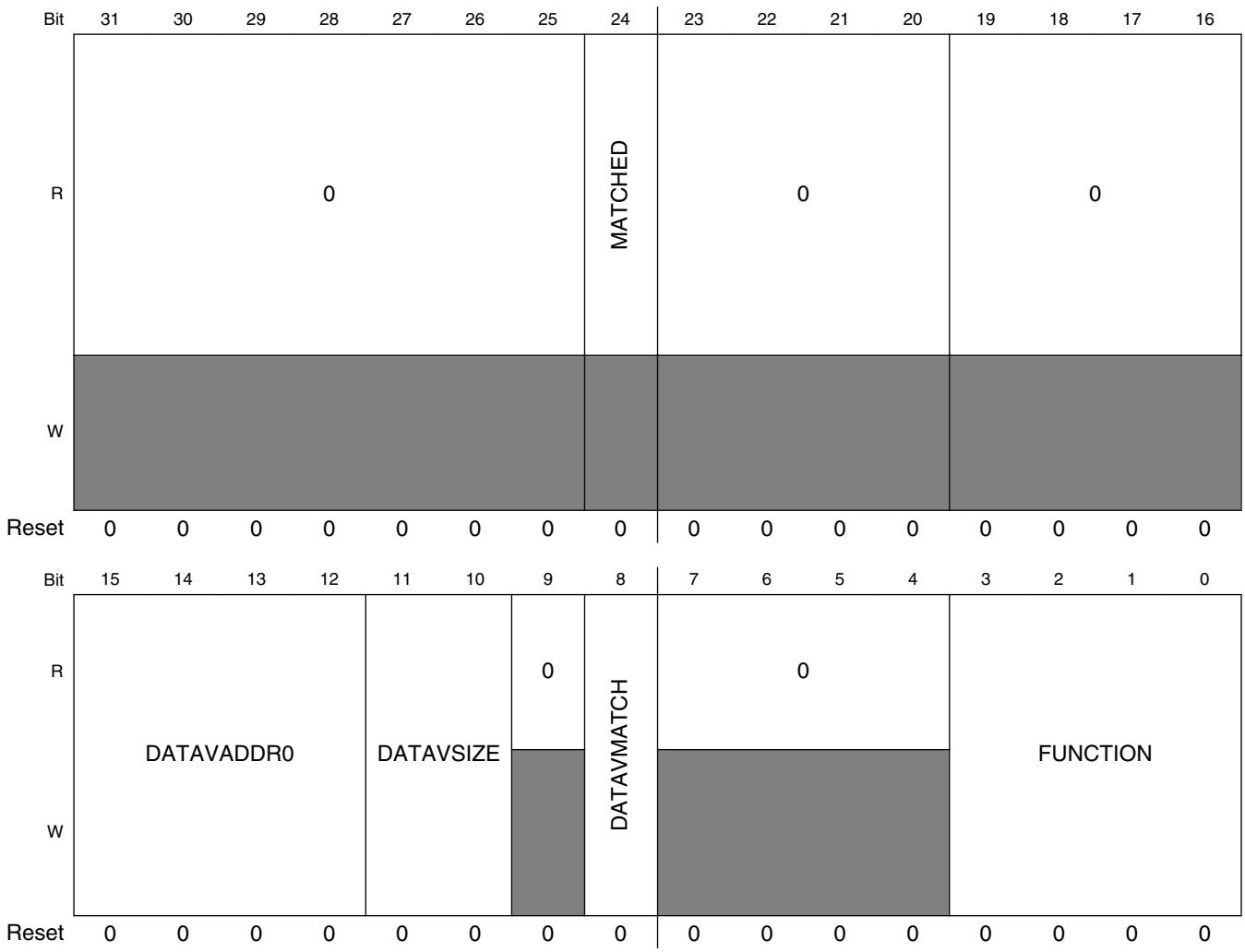
MTBDWT_MASK*n* field descriptions

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.

21.3.2.4 MTB_DWT Comparator Function Register 0 (MTBDWT_FCT0)

The MTBDWT_FCTn registers control the operation of comparator n.

Address: F000_1000h base + 28h offset = F000_1028h



MTBDWT_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...

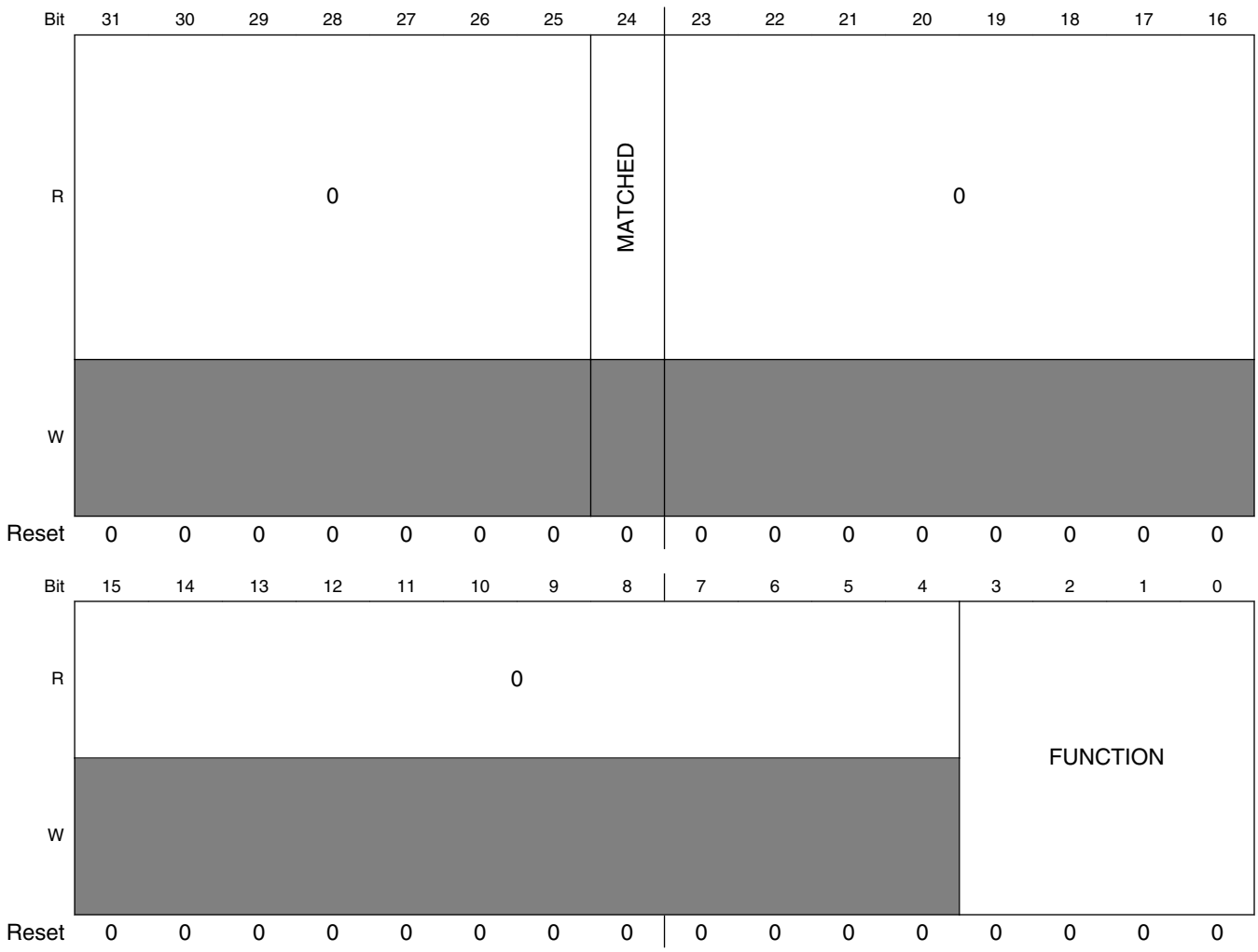
MTBDWT_FCT0 field descriptions (continued)

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.

21.3.2.5 MTB_DWT Comparator Function Register 1 (MTBDWT_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



MTBDWT_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...

MTBDWT_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.

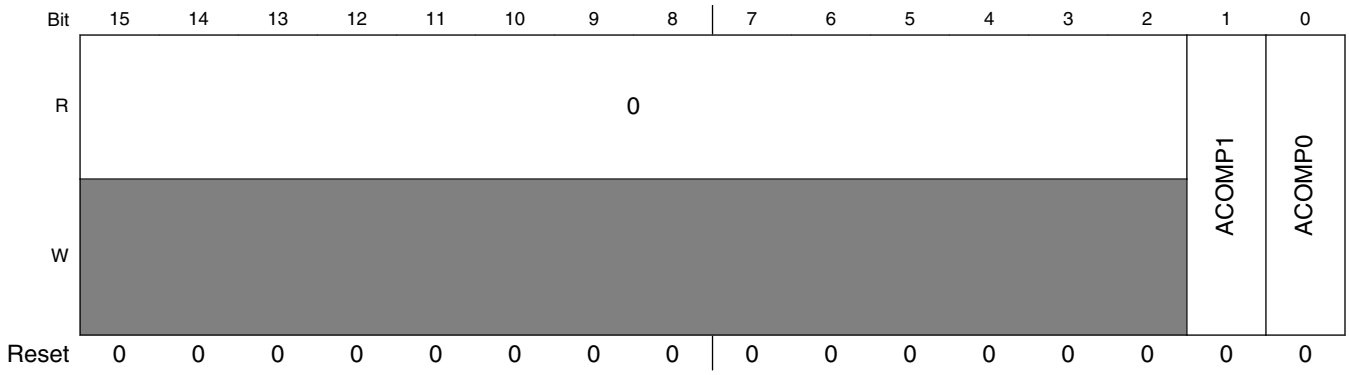
21.3.2.6 MTB_DWT Trace Buffer Control Register (MTBDWT_TBCTRL)

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



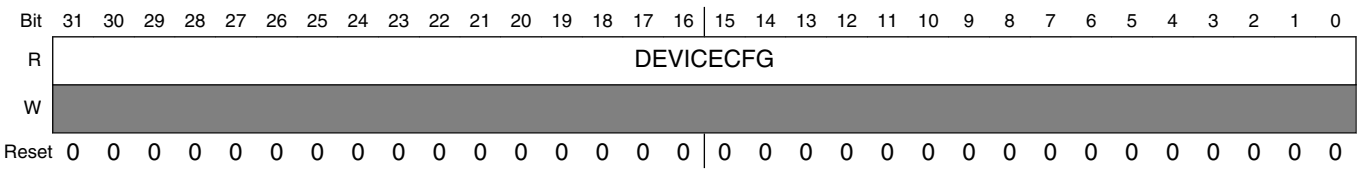
MTBDWT_TBCTRL field descriptions

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].

21.3.2.7 Device Configuration Register (MTBDWT_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



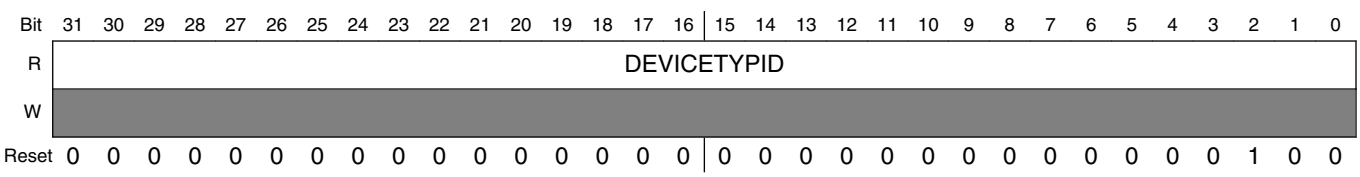
MTBDWT_DEVICECFG field descriptions

Field	Description
DEVICECFG	DEVICECFG Hardwired to 0x0000_0000.

21.3.2.8 Device Type Identifier Register (MTBDWT_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



MTBDWT_DEVICETYPID field descriptions

Field	Description
DEVICETYPID	DEVICETYPID Hardwired to 0x0000_0004.

21.3.2.9 Peripheral ID Register (MTBDWT_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*

MTBDWT_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.

21.3.2.10 Component ID Register (MTBDWT_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*

MTBDWT_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.

21.3.3 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.

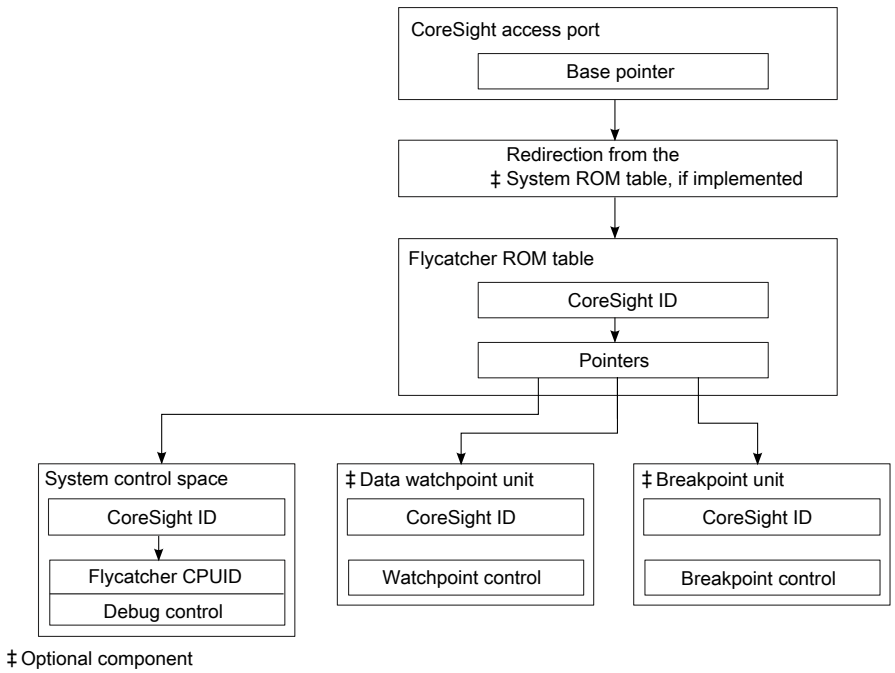


Figure 21-3. CoreSight discovery process

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	21.3.3.1/ 471
F000_2004	Entry (ROM_ENTRY1)	32	R	See section	21.3.3.1/ 471
F000_2008	Entry (ROM_ENTRY2)	32	R	See section	21.3.3.1/ 471
F000_200C	End of Table Marker Register (ROM_TABLEMARK)	32	R	0000_0000h	21.3.3.2/ 472
F000_2FCC	System Access Register (ROM_SYSACCESS)	32	R	0000_0001h	21.3.3.3/ 472
F000_2FD0	Peripheral ID Register (ROM_PERIPHID4)	32	R	See section	21.3.3.4/ 473

Table continues on the next page...

ROM memory map (continued)

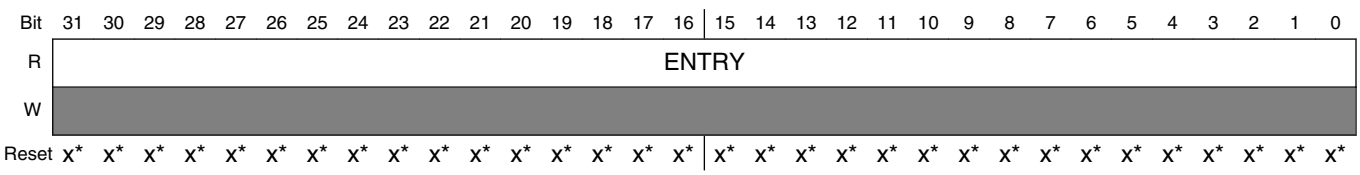
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_2FD4	Peripheral ID Register (ROM_PERIPHID5)	32	R	See section	21.3.3.4/ 473
F000_2FD8	Peripheral ID Register (ROM_PERIPHID6)	32	R	See section	21.3.3.4/ 473
F000_2FDC	Peripheral ID Register (ROM_PERIPHID7)	32	R	See section	21.3.3.4/ 473
F000_2FE0	Peripheral ID Register (ROM_PERIPHID0)	32	R	See section	21.3.3.4/ 473
F000_2FE4	Peripheral ID Register (ROM_PERIPHID1)	32	R	See section	21.3.3.4/ 473
F000_2FE8	Peripheral ID Register (ROM_PERIPHID2)	32	R	See section	21.3.3.4/ 473
F000_2FEC	Peripheral ID Register (ROM_PERIPHID3)	32	R	See section	21.3.3.4/ 473
F000_2FF0	Component ID Register (ROM_COMPID0)	32	R	See section	21.3.3.5/ 473
F000_2FF4	Component ID Register (ROM_COMPID1)	32	R	See section	21.3.3.5/ 473
F000_2FF8	Component ID Register (ROM_COMPID2)	32	R	See section	21.3.3.5/ 473
F000_2FFC	Component ID Register (ROM_COMPID3)	32	R	See section	21.3.3.5/ 473

21.3.3.1 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



ROM_ENTRY*n* field descriptions

Field	Description
ENTRY	ENTRY

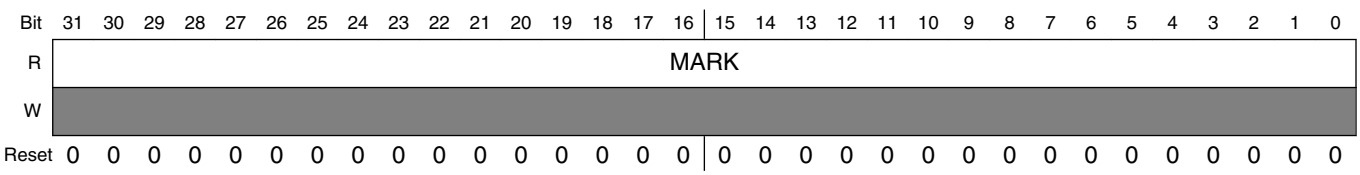
ROM_ENTRY*n* field descriptions (continued)

Field	Description
	Entry 0 (MTB) is hardwired to 0xFFFF_E003; Entry 1 (MTBDWT) to 0xFFFF_F003; Entry 2 (CM0+ ROM Table) to 0xF00F_D003.

21.3.3.2 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



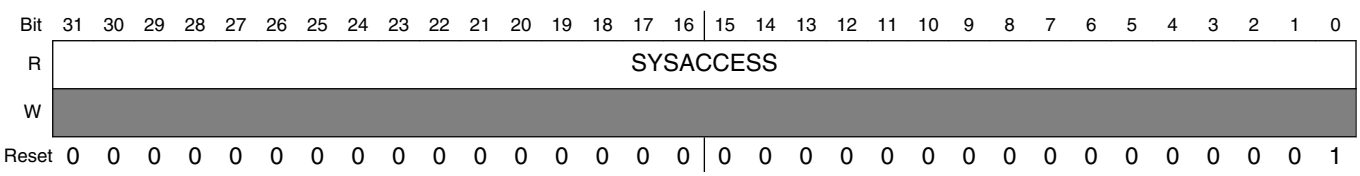
ROM_TABLEMARK field descriptions

Field	Description
MARK	MARK Hardwired to 0x0000_0000

21.3.3.3 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



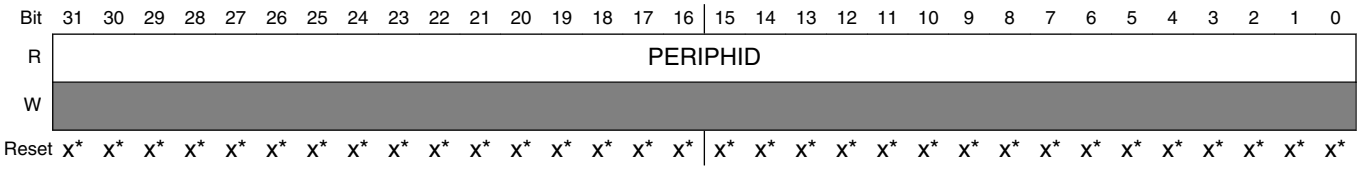
ROM_SYSACCESS field descriptions

Field	Description
SYSACCESS	SYSACCESS Hardwired to 0x0000_0001

21.3.3.4 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



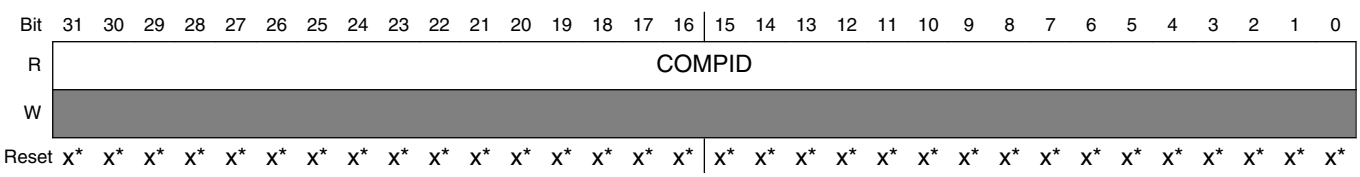
ROM_PERIPHID*n* field descriptions

Field	Description
PERIPHID	PERIPHID Peripheral ID1 is hardwired to 0x0000_00E0; ID2 to 0x0000_0008; and all the others to 0x0000_0000.

21.3.3.5 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



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 22
Crossbar Switch Lite (AXBS-Lite)

22.1 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.

22.1.1 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).

22.2 Functional Description

Information about general operation and arbitration can be found here.

22.2.1 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.

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.

22.2.2 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.

22.2.2.1 Arbitration during undefined length bursts

Undefined length bursts can be interrupted.

22.2.2.2 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 22-1. 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
Both of the following are true: • The current master is running an undefined length burst transfer. • The requesting master's priority level is higher than that of the current master.	At the next arbitration point for the undefined length burst transfer
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

22.2.2.3 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.

22.3 Initialization/application information

No initialization is required for the crossbar switch.

Chapter 23
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.

23.1 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.

23.1.1 Features

Key features of the peripheral bridge are:

- Supports peripheral slots with 8-, 16-, and 32-bit datapath width

23.1.2 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.

23.2 Memory map/register definition

The AIPS module(s) on this device do(es) not contain any user-programmable registers.

23.3 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.

23.3.1 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 24
Direct Memory Access Multiplexer (DMAMUX)

24.1 Introduction

24.1.1 Overview

The Direct Memory Access Multiplexer (DMAMUX) routes DMA sources, called slots, to any of the 4 DMA channels. See the chip-specific information to know the detailed source numbers. This process is illustrated in the following figure.

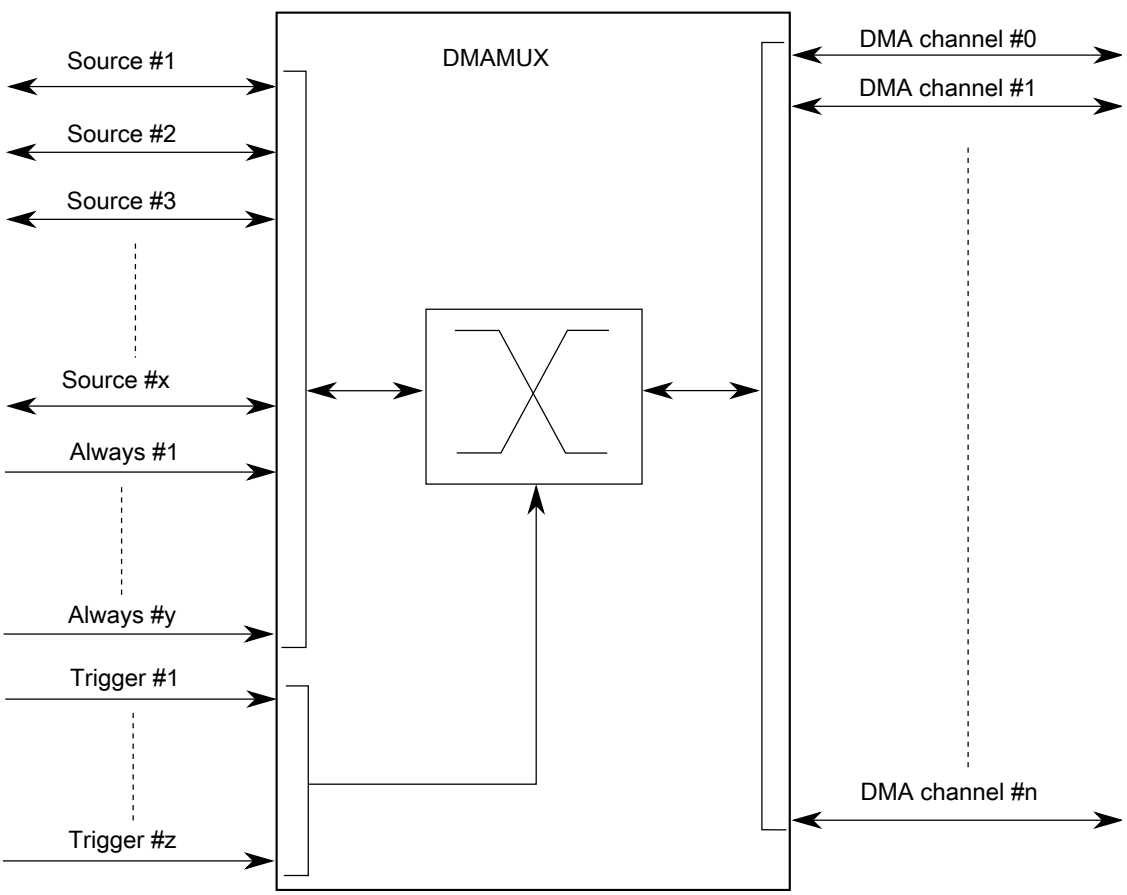


Figure 24-1. DMAMUX block diagram

24.1.2 Features

The DMAMUX module provides these features:

- Up to 59 peripheral slots and up to 4 always-on slots can be routed to 4 channels.
- 4 independently selectable DMA channel routers.
  - The first 2 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.

24.1.3 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.

  Configuration of the period is done by an external periodic interrupt timer (for example, PIT). This mode is available only for channels 0–1.

24.2 External signal description

The DMAMUX has no external pins.

24.3 Memory map/register definition

This section provides a detailed description of all memory-mapped registers in the DMAMUX.

24.3.1 DMAMUX register descriptions

24.3.1.1 DMAMUX Memory map

DMAMUX0 base address: 4002_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h - 3h	Channel Configuration register (CHCFG0 - CHCFG3)	8	RW	00h

24.3.1.2 Channel Configuration register (CHCFG0 - CHCFG3)

24.3.1.2.1 Offset

Register	Offset
CHCFG0	0h
CHCFG1	1h
CHCFG2	2h
CHCFG3	3h

24.3.1.2.2 Function

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].

24.3.1.2.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	ENBL	TRIG	SOURCE					
W								
Reset	0	0	0	0	0	0	0	0

24.3.1.2.4 Fields

Field	Function
7 ENBL	DMA Channel Enable Enables the DMA channel. 0b - 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. 1b - DMA channel is enabled
6 TRIG	DMA Channel Trigger Enable Enables the periodic trigger capability for the triggered DMA channel. 0b - 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) 1b - Triggering is enabled. If triggering is enabled and ENBL is set, the DMAMUX is in Periodic Trigger mode.
5-0 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.

24.4 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

24.4.1 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.

The trigger is generated by an external periodic interrupt timer (for example, PIT); as such, the configuration of the periodic triggering interval is done via configuring the external periodic timer.

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.

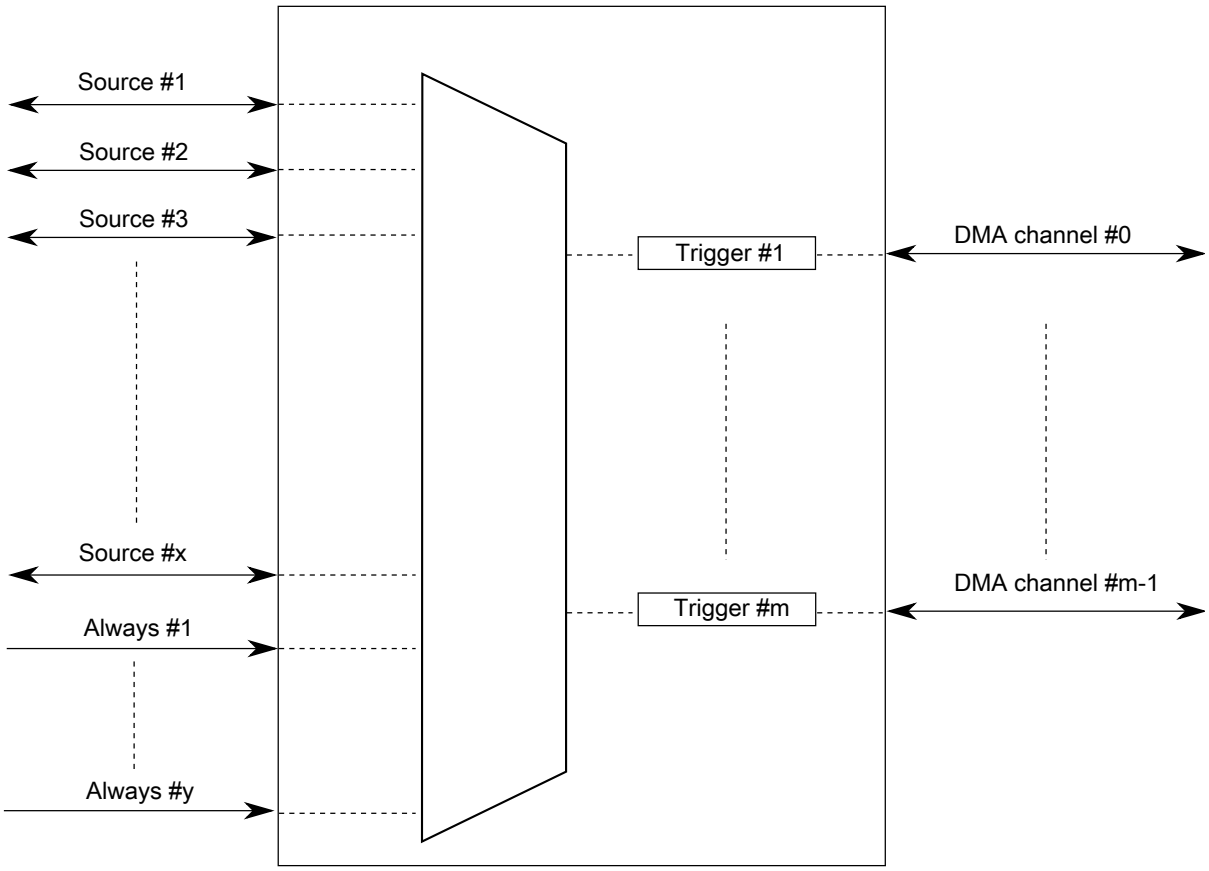


Figure 24-2. DMAMUX triggered channels

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.

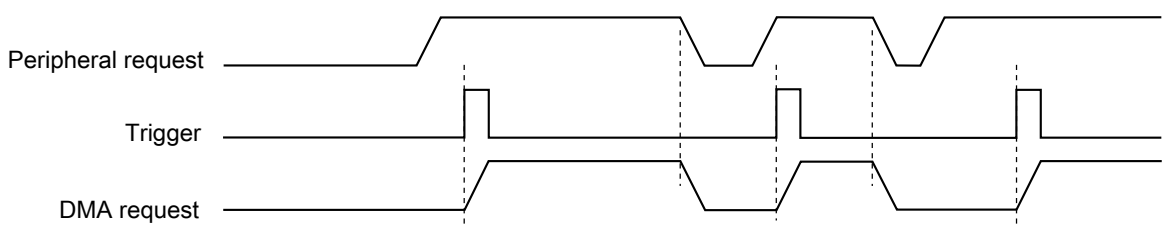


Figure 24-3. DMAMUX channel triggering: normal operation

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.

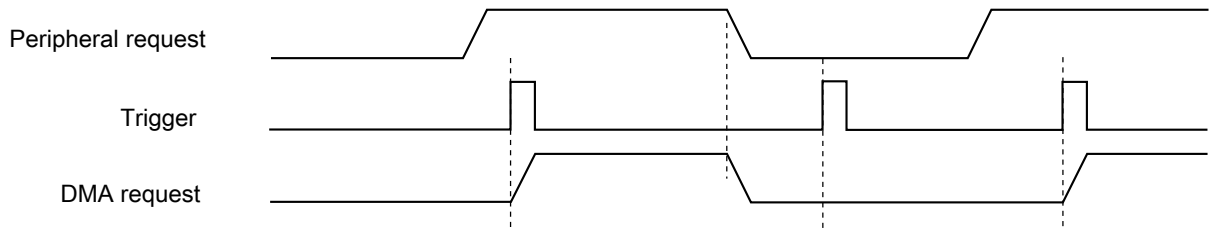


Figure 24-4. DMAMUX channel triggering: ignored trigger

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.

24.4.2 DMA channels with no triggering capability

The other channels of the DMAMUX provide the normal routing functionality as described in Modes of operation.

24.4.3 Always-enabled DMA sources

In addition to the peripherals that can be used as DMA sources, there are 4 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.

24.5 Initialization/application information

This section provides instructions for initializing the DMA channel MUX.

24.5.1 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.

24.5.2 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-specific information 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 setwhile 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:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0x40021000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCFG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCFG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCFG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCFG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCFG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCFG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCFG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCFG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCFG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCFG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCFG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCFG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCFG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCFG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
```

```
volatile unsigned char *CHCFG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCFG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);

In File main.c:
#include "registers.h"
:
:
*CHCFG1 = 0x00;
*CHCFG1 = 0x85;
```

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:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0x40021000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCFG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCFG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCFG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCFG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCFG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCFG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCFG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCFG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCFG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCFG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCFG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCFG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCFG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCFG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCFG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCFG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);
```

```
In File main.c:
#include "registers.h"
:
:
*CHCFG8 = 0x00;
*CHCFG8 = 0x87;
```

Chapter 25
DMA Controller Module

25.1 Introduction

The enhanced direct memory access (eDMA) controller is a second-generation module capable of performing complex data transfers with minimal intervention from a host processor. The hardware microarchitecture includes:

- A DMA engine that performs:
  - Source address and destination address calculations
  - Data-movement operations
- Local memory containing transfer control descriptors for each of the 4 channels

25.1.1 eDMA system block diagram

Figure 25-1 illustrates the components of the eDMA system, including the eDMA module ("engine").

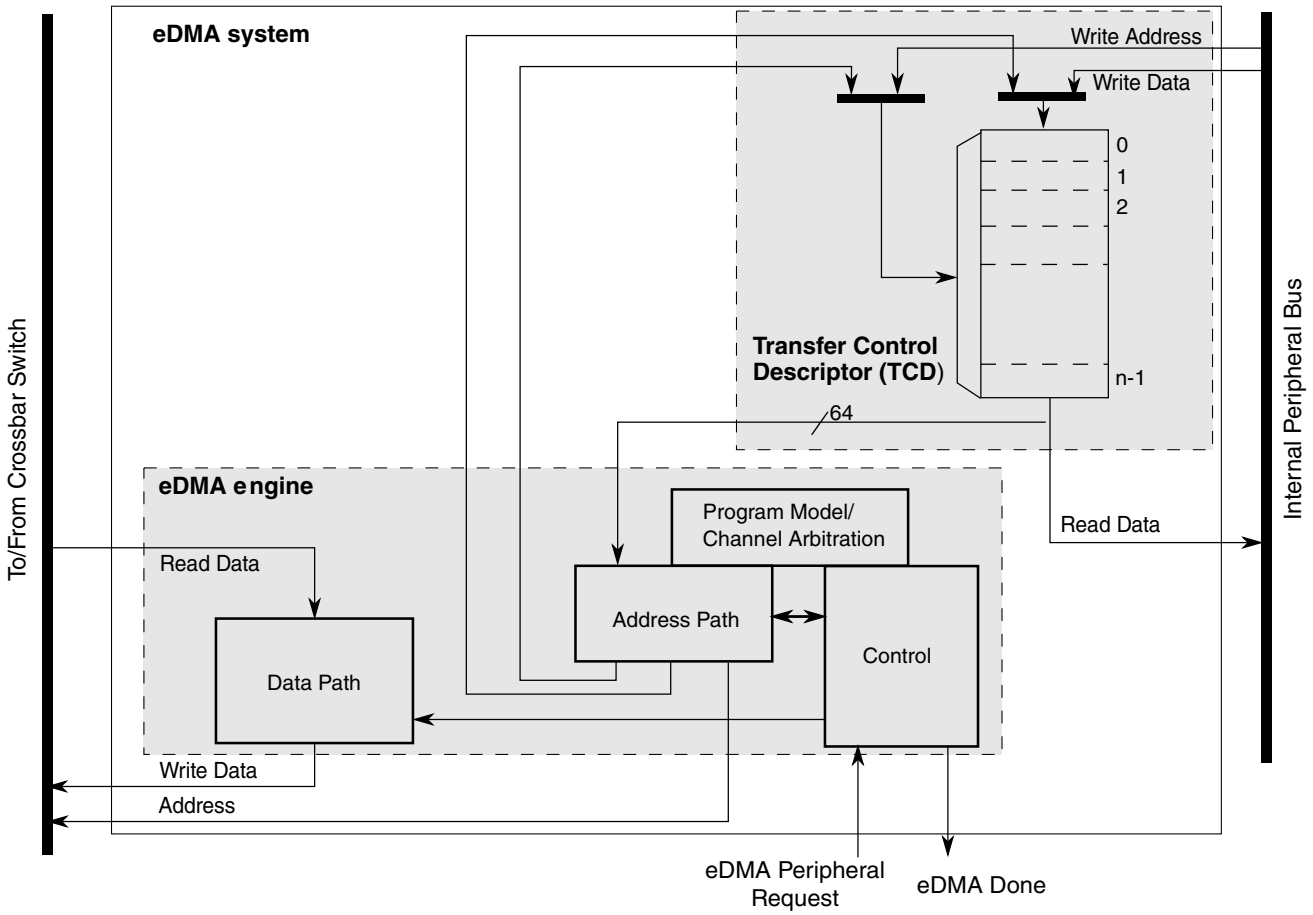


Figure 25-1. eDMA system block diagram

25.1.2 Block parts

The eDMA module is partitioned into two major modules: the eDMA engine and the transfer-control descriptor local memory.

The eDMA engine is further partitioned into four submodules:

Table 25-1. eDMA engine submodules

Submodule	Function
Address path	This block implements registered versions of two channel transfer control descriptors, channel x and channel y, and manages all master bus-address calculations. All the channels provide the same functionality. This structure allows data transfers associated with one channel to be preempted after the completion of a read/write sequence if a higher priority channel activation is asserted while the first channel is active. After a channel is activated, it runs until the minor loop is completed, unless preempted by a higher priority channel. This provides a mechanism (enabled by DCHPRI*n*[ECP]) where a large data move operation can be preempted to minimize the time another channel is blocked from execution. When any channel is selected to execute, the contents of its TCD are read from local memory and loaded into the address path channel x registers for a normal start and into channel y registers for a preemption start. After the minor loop completes execution, the address path hardware writes

Table continues on the next page...

Table 25-1. eDMA engine submodules (continued)

Submodule	Function
	the new values for the TCD*n*_{SADDR, DADDR, CITER} back to local memory. If the major iteration count is exhausted, additional processing is performed, including the final address pointer updates, reloading the TCD*n*_CITER field, and a possible fetch of the next TCD*n* from memory as part of a scatter/gather operation.
Data path	This block implements the bus master read/write datapath. It includes a data buffer and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output. The address and data path modules directly support the 2-stage pipelined internal bus. The address path module represents the 1st stage of the bus pipeline (address phase), while the data path module implements the 2nd stage of the pipeline (data phase).
Program model/channel arbitration	This block implements the first section of the eDMA programming model as well as the channel arbitration logic. The programming model registers are connected to the internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs are also connected to this block (via control logic).
Control	This block provides all the control functions for the eDMA engine. For data transfers where the source and destination sizes are equal, the eDMA engine performs a series of source read/ destination write operations until the number of bytes specified in the minor loop byte count has moved. For descriptors where the sizes are not equal, multiple accesses of the smaller size data are required for each reference of the larger size. As an example, if the source size references 16-bit data and the destination is 32-bit data, two reads are performed, then one 32-bit write.

The transfer-control descriptor local memory is further partitioned into:

Table 25-2. Transfer control descriptor memory

Submodule	Description
Memory controller	This logic implements the required dual-ported controller, managing accesses from the eDMA engine as well as references from the internal peripheral bus. As noted earlier, in the event of simultaneous accesses, the eDMA engine is given priority and the peripheral transaction is stalled.
Memory array	TCD storage for each channel's transfer profile.

25.1.3 Features

The eDMA is a highly programmable data-transfer engine optimized to minimize any required intervention from the host processor. It is intended for use in applications where the data size to be transferred is statically known and not defined within the transferred data itself. The eDMA module features:

- All data movement via dual-address transfers: read from source, write to destination
  - Programmable source and destination addresses and transfer size
  - Support for enhanced addressing modes

- 4-channel implementation that performs complex data transfers with minimal intervention from a host processor
  - Connections to the crossbar switch for bus mastering the data movement
- Transfer control descriptor (TCD) organized to support two-deep, nested transfer operations
  - 32-byte TCD stored in local memory for each channel
  - An inner data transfer loop defined by a minor byte transfer count
  - An outer data transfer loop defined by a major iteration count
- Channel activation via one of three methods:
  - Explicit software initiation
  - Initiation via a channel-to-channel linking mechanism for continuous transfers
  - Peripheral-paced hardware requests, one per channel
- Fixed-priority and round-robin channel arbitration
- Channel completion reported via programmable interrupt requests
  - One interrupt per channel, which can be asserted at completion of major iteration count
  - Programmable error terminations per channel and logically summed together to form one error interrupt to the interrupt controller
- Programmable support for scatter/gather DMA processing
- Support for complex data structures

In the discussion of this module, *n* is used to reference the channel number.

25.2 Modes of operation

The eDMA operates in the following modes:

Table 25-3. Modes of operation

Mode	Description
Normal	In Normal mode, the eDMA transfers data between a source and a destination. The source and destination can be a memory block or an I/O block capable of operation with the eDMA.

Table continues on the next page...

Table 25-3. Modes of operation (continued)

Mode	Description
	A service request initiates a transfer of a specific number of bytes (NBYTES) as specified in the transfer control descriptor (TCD). The minor loop is the sequence of read-write operations that transfers these NBYTES per service request. Each service request executes one iteration of the major loop, which transfers NBYTES of data.
Debug	DMA operation is configurable in Debug mode via the control register: • If CR[EDBG] is cleared, the DMA continues to operate. • If CR[EDBG] is set, the eDMA stops transferring data. If Debug mode is entered while a channel is active, the eDMA continues operation until the channel retires.
Wait	Before entering Wait mode, the DMA attempts to complete its current transfer. After the transfer completes, the device enters Wait mode.

25.3 Memory map/register definition

The eDMA's programming model is partitioned into two regions:

- The first region defines a number of registers providing control functions
- The second region corresponds to the local transfer control descriptor (TCD) memory

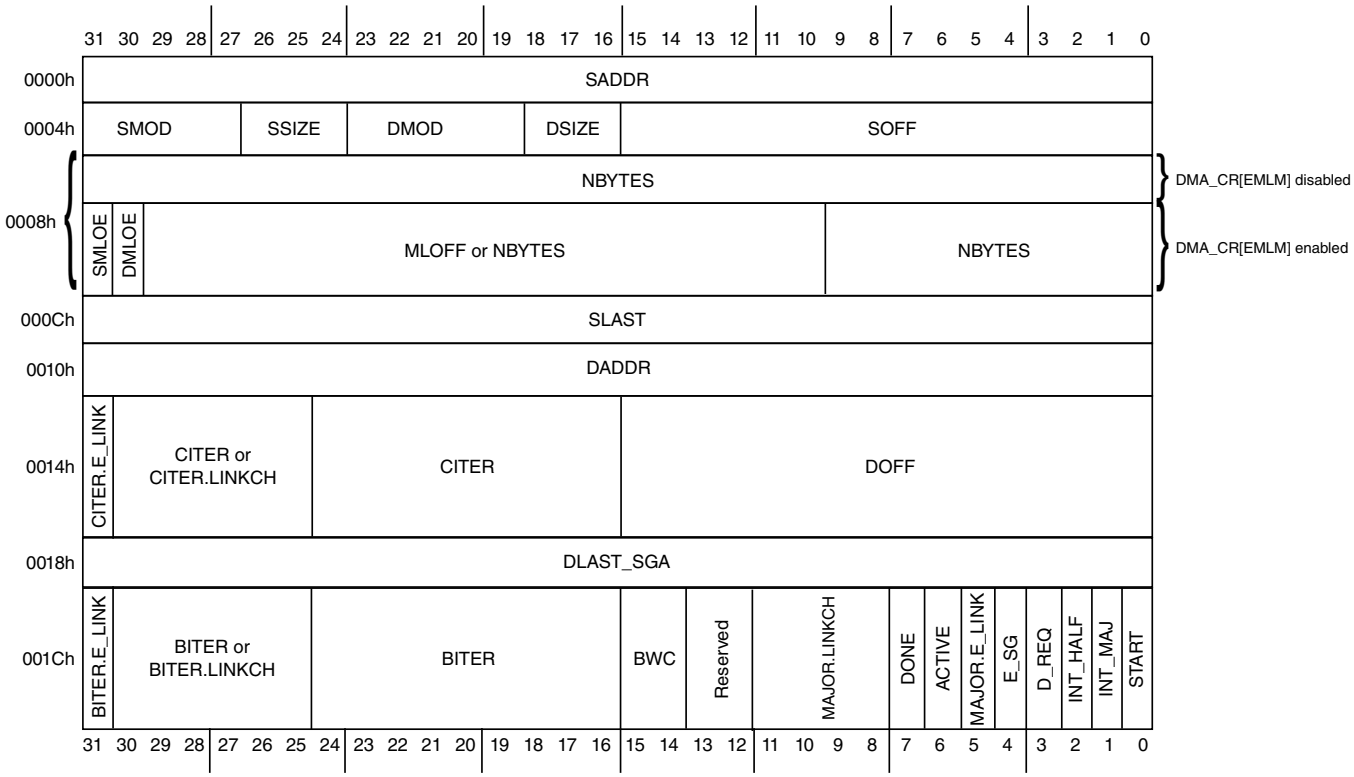
25.3.1 TCD memory

Each channel requires a 32-byte transfer control descriptor for defining the desired data movement operation. The channel descriptors are stored in the local memory in sequential order: channel 0, channel 1, ... channel 3. Each TCD*n* definition is presented as 11 registers of 16 or 32 bits.

25.3.2 TCD initialization

Prior to activating a channel, you must initialize its TCD with the appropriate transfer profile.

25.3.3 TCD structure



25.3.4 Reserved memory and bit fields

- Reading reserved bits in a register returns the value of zero.
- Writes to reserved bits in a register are ignored.
- Reading or writing a reserved memory location generates a bus error.

25.3.5 DMA register descriptions

25.3.5.1 DMA Memory map

DMA0 base address: 4000_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Control Register (CR)	32	RW	See description.

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4h	Error Status Register (ES)	32	RO	0000_0000h
Ch	Enable Request Register (ERQ)	32	RW	0000_0000h
14h	Enable Error Interrupt Register (EEI)	32	RW	0000_0000h
18h	Clear Enable Error Interrupt Register (CEEI)	8	WORZ	00h
19h	Set Enable Error Interrupt Register (SEEI)	8	WORZ	00h
1Ah	Clear Enable Request Register (CERQ)	8	WORZ	00h
1Bh	Set Enable Request Register (SERQ)	8	WORZ	00h
1Ch	Clear DONE Status Bit Register (CDNE)	8	WORZ	00h
1Dh	Set START Bit Register (SSRT)	8	WORZ	00h
1Eh	Clear Error Register (CERR)	8	WORZ	00h
1Fh	Clear Interrupt Request Register (CINT)	8	WORZ	00h
24h	Interrupt Request Register (INT)	32	W1C	0000_0000h
2Ch	Error Register (ERR)	32	W1C	0000_0000h
34h	Hardware Request Status Register (HRS)	32	RO	0000_0000h
44h	Enable Asynchronous Request in Stop Register (EARS)	32	RW	0000_0000h
100h	Channel Priority Register (DCHPRI3)	8	RW	03h
101h	Channel Priority Register (DCHPRI2)	8	RW	02h
102h	Channel Priority Register (DCHPRI1)	8	RW	01h
103h	Channel Priority Register (DCHPRI0)	8	RW	00h
1000h - 1060h	TCD Source Address (TCD0_SADDR - TCD3_SADDR)	32	RW	See description.
1004h - 1064h	TCD Signed Source Address Offset (TCD0_SOFF - TCD3_SOFF)	16	RW	See description.
1006h - 1066h	TCD Transfer Attributes (TCD0_ATTR - TCD3_ATTR)	16	RW	See description.
1008h - 1068h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD0_NBYTES_MLNO - TCD3_NBYTES_MLNO)	32	RW	See description.
1008h - 1068h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD0_NBYTES_MLOFFNO - TCD3_NBYTES_MLOFFNO)	32	RW	See description.
1008h - 1068h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD0_NBYTES_MLOFFYES - TCD3_NBYTES_MLOFFYES)	32	RW	See description.
100Ch - 106Ch	TCD Last Source Address Adjustment (TCD0_SLAST - TCD3_SLAST)	32	RW	See description.
1010h - 1070h	TCD Destination Address (TCD0_DADDR - TCD3_DADDR)	32	RW	See description.
1014h - 1074h	TCD Signed Destination Address Offset (TCD0_DOFF - TCD3_DOFF)	16	RW	See description.
1016h - 1076h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD3_CITER_ELINKNO)	16	RW	See description.
1016h - 1076h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD3_CITER_ELINKYES)	16	RW	See description.

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1018h - 1078h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD0_DLASTSGA - TCD3_DLASTSGA)	32	RW	See description.
101Ch - 107Ch	TCD Control and Status (TCD0_CSR - TCD3_CSR)	16	RW	See description.
101Eh - 107Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD3_BITER_ELINKNO)	16	RW	See description.
101Eh - 107Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD3_BITER_ELINKYES)	16	RW	See description.

25.3.5.2 Control Register (CR)

25.3.5.2.1 Offset

Register	Offset
CR	0h

25.3.5.2.2 Function

The CR defines the basic operating configuration of the DMA.

Arbitration can be configured to use either a fixed-priority or a round-robin scheme. For fixed-priority arbitration, the highest priority channel requesting service is selected to execute. The channel priority registers assign the priorities; see the DCHPRIn registers. For round-robin arbitration, the channel priorities are ignored and channels are cycled through (from high to low channel number) without regard to priority.

NOTE

For correct operation, writes to the CR register must be performed only when the DMA channels are inactive; that is, when TCDn_CSR[ACTIVE] bits are cleared.

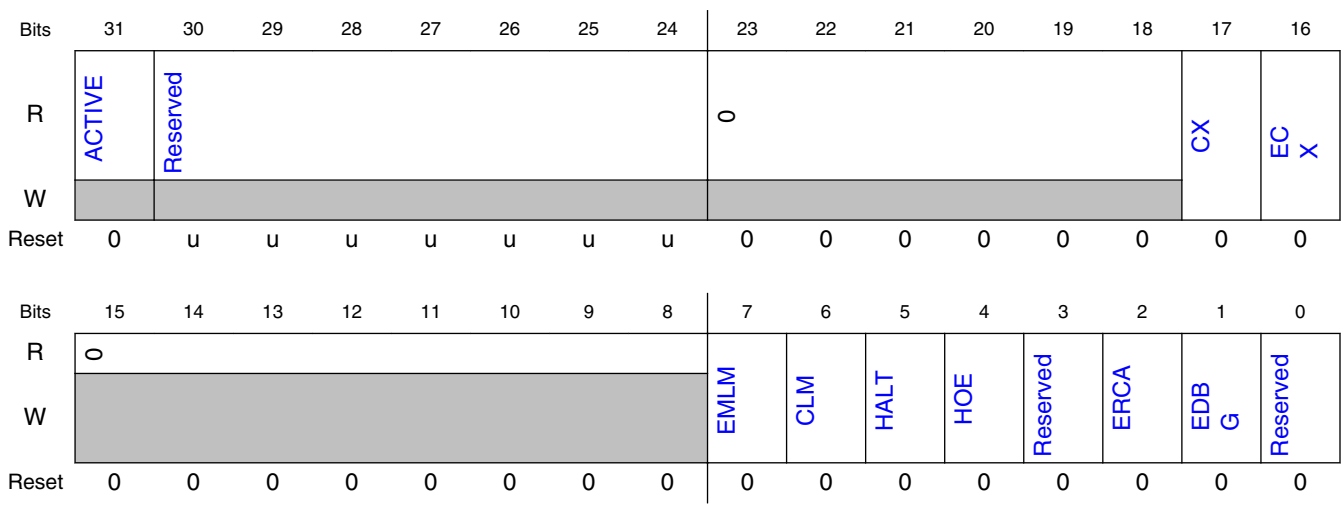
Minor loop offsets are address offset values added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) upon minor loop completion. When minor loop offsets are enabled, the minor loop offset (MLOFF) is added to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both prior to the addresses being written back into the TCD. If the major loop is

complete, the minor loop offset is ignored and the major loop address offsets (TCDn_SLAST and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When minor loop mapping is enabled (EMLM is 1), TCDn word2 is redefined. A portion of TCDn word2 is used to specify multiple fields: a source enable bit (SMLOE) to specify the minor loop offset should be applied to the source address (TCDn_SADDR) upon minor loop completion, a destination enable bit (DMLOE) to specify the minor loop offset should be applied to the destination address (TCDn_DADDR) upon minor loop completion, and the sign extended minor loop offset value (MLOFF). The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. When both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

When minor loop mapping is disabled (EMLM is 0), all 32 bits of TCDn word2 are assigned to the NBYTES field.

25.3.5.2.3 Diagram



25.3.5.2.4 Fields

Field	Function
31 ACTIVE	DMA Active Status 0b - eDMA is idle. 1b - eDMA is executing a channel.
30-24 —	eDMA version number Reserved

Table continues on the next page...

Field	Function
23-18 —	Reserved
17 CX	Cancel Transfer 0b - Normal operation 1b - Cancel the remaining data transfer. Stop the executing channel and force the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. The CX bit clears itself after the cancel has been honored. This cancel retires the channel normally as if the minor loop was completed.
16 ECX	Error Cancel Transfer 0b - Normal operation 1b - Cancel the remaining data transfer in the same fashion as the CX bit. Stop the executing channel and force the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. The ECX bit clears itself after the cancel is honored. In addition to cancelling the transfer, ECX treats the cancel as an error condition, thus updating the Error Status register (DMAx_ES) and generating an optional error interrupt.
15-8 —	Reserved
7 EMLM	Enable Minor Loop Mapping 0b - Disabled. TCDn.word2 is defined as a 32-bit NBYTES field. 1b - Enabled. TCDn.word2 is redefined to include individual enable fields, an offset field, and the NBYTES field. The individual enable fields allow the minor loop offset to be applied to the source address, the destination address, or both. The NBYTES field is reduced when either offset is enabled.
6 CLM	Continuous Link Mode **NOTE:** Do not use continuous link mode with a channel linking to itself if there is only one minor loop iteration per service request, for example, if the channel's NBYTES value is the same as either the source or destination size. The same data transfer profile can be achieved by simply increasing the NBYTES value, which provides more efficient, faster processing. 0b - A minor loop channel link made to itself goes through channel arbitration before being activated again. 1b - A minor loop channel link made to itself does not go through channel arbitration before being activated again. Upon minor loop completion, the channel activates again if that channel has a minor loop channel link enabled and the link channel is itself. This effectively applies the minor loop offsets and restarts the next minor loop.
5 HALT	Halt DMA Operations 0b - Normal operation 1b - Stall the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this bit is cleared.
4 HOE	Halt On Error 0b - Normal operation 1b - Any error causes the HALT bit to set. Subsequently, all service requests are ignored until the HALT bit is cleared.
3 —	Reserved Reserved
2 ERCA	Enable Round Robin Channel Arbitration 0b - Fixed priority arbitration is used for channel selection . 1b - Round robin arbitration is used for channel selection .
1 EDBG	Enable Debug 0b - When in debug mode, the DMA continues to operate.

Table continues on the next page...

Field	Function
	1b - When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. Channel execution resumes when the system exits debug mode or the EDBG bit is cleared.
0	Reserved
—	Reserved

25.3.5.3 Error Status Register (ES)

25.3.5.3.1 Offset

Register	Offset
ES	4h

25.3.5.3.2 Function

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- A configuration error, that is:
  - An illegal setting in the transfer-control descriptor, or
  - An illegal priority register setting in fixed-arbitration
- An error termination to a bus master read or write cycle
- A cancel transfer with error bit that will be set when a transfer is canceled via the corresponding cancel transfer control bit

See Fault reporting and handling for more details.

25.3.5.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	0														ECX
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	0	CPE	0				ERRCHN		SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.5.3.4 Fields

Field	Function
31 VLD	VLD Logical OR of all ERR status bits 0b - No ERR bits are set. 1b - At least one ERR bit is set indicating a valid error exists that has not been cleared.
30-17 —	Reserved
16 ECX	Transfer Canceled 0b - No canceled transfers 1b - The last recorded entry was a canceled transfer by the error cancel transfer input
15 —	Reserved
14 CPE	Channel Priority Error 0b - No channel priority error 1b - The last recorded error was a configuration error in the channel priorities . Channel priorities are not unique.
13-10 —	Reserved
9-8 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the last recorded error, excluding CPE errors, or last recorded error canceled transfer.
7 SAE	Source Address Error 0b - No source address configuration error. 1b - The last recorded error was a configuration error detected in the TCDn_SADDR field. TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].
6 SOE	Source Offset Error 0b - No source offset configuration error 1b - The last recorded error was a configuration error detected in the TCDn_SOFF field. TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE].
5 DAE	Destination Address Error 0b - No destination address configuration error 1b - The last recorded error was a configuration error detected in the TCDn_DADDR field. TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE].
4 DOE	Destination Offset Error 0b - No destination offset configuration error 1b - The last recorded error was a configuration error detected in the TCDn_DOFF field. TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE].
3 NCE	NBYTES/CITER Configuration Error 0b - No NBYTES/CITER configuration error 1b - The last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields. TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE], orTCDn_CITER[CITER] is equal to zero, orTCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK]
2 SGE	Scatter/Gather Configuration Error 0b - No scatter/gather configuration error

Table continues on the next page...

Field	Function
	1b - The last recorded error was a configuration error detected in the TCDn_DLASTSGA field. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. TCDn_DLASTSGA is not on a 32 byte boundary.
1 SBE	Source Bus Error 0b - No source bus error 1b - The last recorded error was a bus error on a source read
0 DBE	Destination Bus Error 0b - No destination bus error 1b - The last recorded error was a bus error on a destination write

25.3.5.4 Enable Request Register (ERQ)

25.3.5.4.1 Offset

Register	Offset
ERQ	Ch

25.3.5.4.2 Function

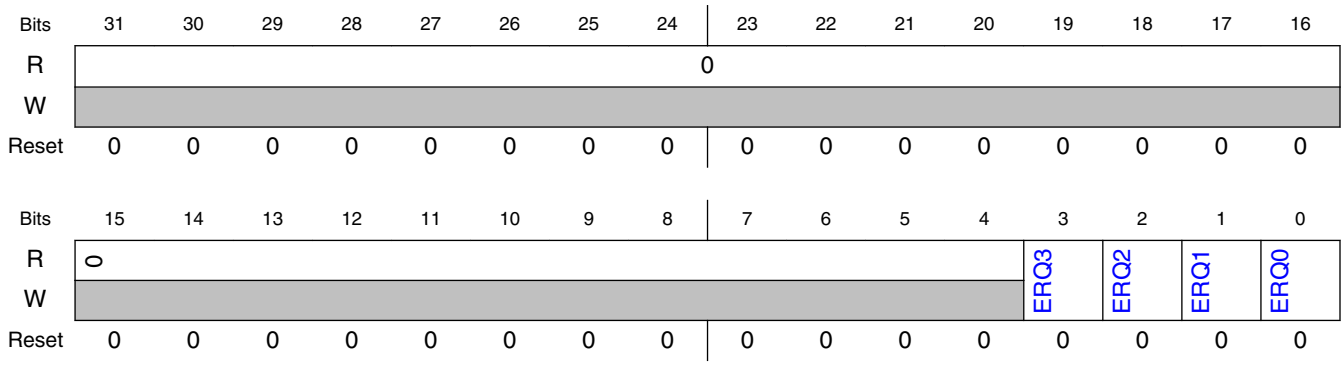
The ERQ register provides a bit map for the 4 channels to enable the request signal for each channel. The state of any given channel enable is directly affected by writes to this register; it is also affected by writes to the SERQ and CERQ registers. These registers are provided so the request enable for a single channel can easily be modified without needing to perform a read-modify-write sequence to this register.

DMA request input signals and this enable request flag must be asserted before a channel's hardware service request is accepted. The state of the DMA enable request flag does not affect a channel service request made explicitly through software or a linked channel request.

NOTE

Disable a channel's hardware service request at the source before clearing the channel's ERQ bit.

25.3.5.4.3 Diagram



25.3.5.4.4 Fields

Field	Function
31-4 —	Reserved
3 ERQ3	Enable DMA Request 3 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
2 ERQ2	Enable DMA Request 2 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
1 ERQ1	Enable DMA Request 1 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
0 ERQ0	Enable DMA Request 0 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled

25.3.5.5 Enable Error Interrupt Register (EEI)

25.3.5.5.1 Offset

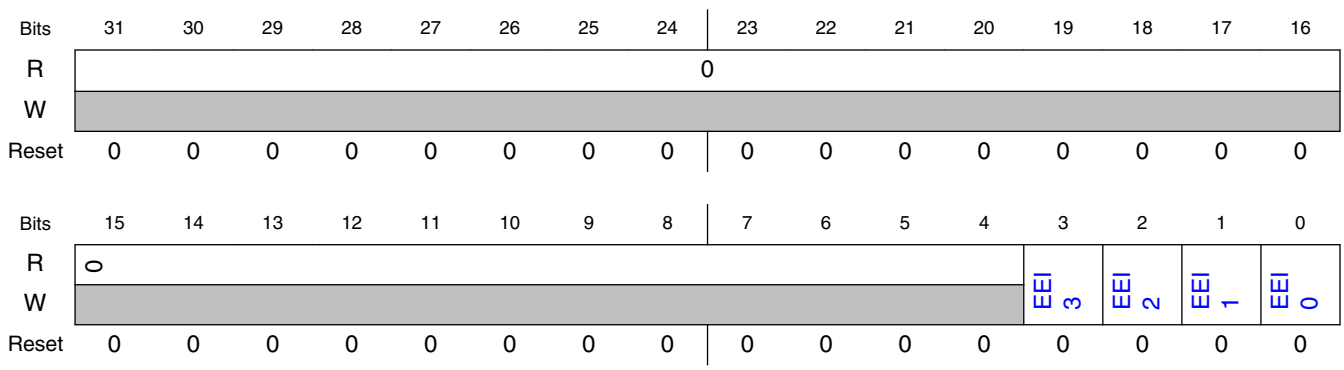
Register	Offset
EEI	14h

25.3.5.5.2 Function

The EEI register provides a bit map for the 4 channels to enable the error interrupt signal for each channel. The state of any given channel's error interrupt enable is directly affected by writes to this register; it is also affected by writes to the SEEI and CEEI. These registers are provided so that the error interrupt enable for a single channel can easily be modified without the need to perform a read-modify-write sequence to the EEI register.

The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller.

25.3.5.5.3 Diagram



25.3.5.5.4 Fields

Field	Function
31-4 —	Reserved
3 EEI3	Enable Error Interrupt 3 0b - The error signal for corresponding channel does not generate an error interrupt 1b - The assertion of the error signal for corresponding channel generates an error interrupt request
2 EEI2	Enable Error Interrupt 2 0b - The error signal for corresponding channel does not generate an error interrupt 1b - The assertion of the error signal for corresponding channel generates an error interrupt request
1 EEI1	Enable Error Interrupt 1 0b - The error signal for corresponding channel does not generate an error interrupt 1b - The assertion of the error signal for corresponding channel generates an error interrupt request
0 EEI0	Enable Error Interrupt 0 0b - The error signal for corresponding channel does not generate an error interrupt 1b - The assertion of the error signal for corresponding channel generates an error interrupt request

25.3.5.6 Clear Enable Error Interrupt Register (CEEI)

25.3.5.6.1 Offset

Register	Offset
CEEI	18h

25.3.5.6.2 Function

The CEEI provides a simple memory-mapped mechanism to clear a given bit in the EEI to disable the error interrupt for a given channel. The data value on a register write causes the corresponding bit in the EEI to be cleared. Setting the CAEE bit provides a global clear function, forcing the EEI contents to be cleared, disabling all DMA request inputs.

If the NOP bit is set, the command is ignored. This allows you to set a single, byte-wide register with a 32-bit write while not affecting the other registers addressed in the write. In such a case the other three bytes of the word would all have their NOP bit set so that that these register will not be affected by the write.

Reads of this register return all zeroes.

25.3.5.6.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	0	0					0	
W	NOP	CAEE	0				CEEI	
Reset	0	0	0	0	0	0	0	0

25.3.5.6.4 Fields

Field	Function
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register
6 CAEE	Clear All Enable Error Interrupts 0b - Clear only the EEI bit specified in the CEEI field 1b - Clear all bits in EEI
5-2 —	Reserved

Table continues on the next page...

Field	Function
1-0 CEEI	Clear Enable Error Interrupt Clears the corresponding bit in EEI

25.3.5.7 Set Enable Error Interrupt Register (SEEI)

25.3.5.7.1 Offset

Register	Offset
SEEI	19h

25.3.5.7.2 Function

The SEEI provides a simple memory-mapped mechanism to set a given bit in the EEI to enable the error interrupt for a given channel. The data value on a register write causes the corresponding bit in the EEI to be set. Setting the SAEE bit provides a global set function, forcing the entire EEI contents to be set.

If the NOP bit is set, the command is ignored. This allows you to set a single, byte-wide register with a 32-bit write while not affecting the other registers addressed in the write. In such a case the other three bytes of the word would all have their NOP bit set so that that these register will not be affected by the write.

Reads of this register return all zeroes.

25.3.5.7.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	0	0					0	
W	NOP	SAEE	0				SEEI	
Reset	0	0	0	0	0	0	0	0

25.3.5.7.4 Fields

Field	Function
7	No Op enable 0b - Normal operation

Table continues on the next page...

Field	Function
NOP	1b - No operation, ignore the other bits in this register
6 SAEE	Sets All Enable Error Interrupts 0b - Set only the EEI bit specified in the SEEI field. 1b - Sets all bits in EEI
5-2 —	Reserved
1-0 SEEI	Set Enable Error Interrupt Sets the corresponding bit in EEI

25.3.5.8 Clear Enable Request Register (CERQ)

25.3.5.8.1 Offset

Register	Offset
CERQ	1Ah

25.3.5.8.2 Function

The CERQ provides a simple memory-mapped mechanism to clear a given bit in the ERQ to disable the DMA request for a given channel. The data value on a register write causes the corresponding bit in the ERQ to be cleared. Setting the CAER bit provides a global clear function, forcing the entire contents of the ERQ to be cleared, disabling all DMA request inputs.

If the NOP bit is set, the command is ignored. This allows you to set a single, byte-wide register with a 32-bit write while not affecting the other registers addressed in the write. In such a case the other three bytes of the word would all have their NOP bit set so that that these register will not be affected by the write.

Reads of this register return all zeroes.

NOTE

Disable a channel's hardware service request at the source before clearing the channel's ERQ bit.

25.3.5.8.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	0	0					0	
W	NOP	CAER	0				CERQ	
Reset	0	0	0	0	0	0	0	0

25.3.5.8.4 Fields

Field	Function
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register
6 CAER	Clear All Enable Requests 0b - Clear only the ERQ bit specified in the CERQ field 1b - Clear all bits in ERQ
5-2 —	Reserved
1-0 CERQ	Clear Enable Request Clears the corresponding bit in ERQ.

25.3.5.9 Set Enable Request Register (SERQ)

25.3.5.9.1 Offset

Register	Offset
SERQ	1Bh

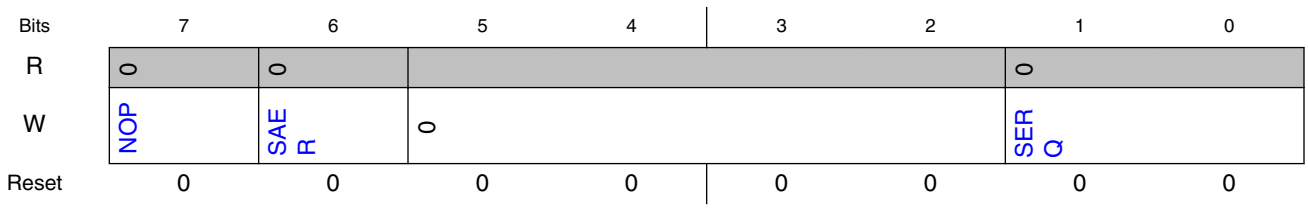
25.3.5.9.2 Function

The SERQ provides a simple memory-mapped mechanism to set a given bit in the ERQ to enable the DMA request for a given channel. The data value on a register write causes the corresponding bit in the ERQ to be set. Setting the SAER bit provides a global set function, forcing the entire contents of ERQ to be set.

If the NOP bit is set, the command is ignored. This allows you to set a single, byte-wide register with a 32-bit write while not affecting the other registers addressed in the write. In such a case the other three bytes of the word would all have their NOP bit set so that that these register will not be affected by the write.

Reads of this register return all zeroes.

25.3.5.9.3 Diagram



25.3.5.9.4 Fields

Field	Function
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register
6 SAER	Set All Enable Requests 0b - Set only the ERQ bit specified in the SERQ field 1b - Set all bits in ERQ
5-2 —	Reserved
1-0 SERQ	Set Enable Request Sets the corresponding bit in ERQ.

25.3.5.10 Clear DONE Status Bit Register (CDNE)

25.3.5.10.1 Offset

Register	Offset
CDNE	1Ch

25.3.5.10.2 Function

The CDNE provides a simple memory-mapped mechanism to clear the DONE bit in the TCD of the given channel. The data value on a register write causes the DONE bit in the corresponding transfer control descriptor to be cleared. Setting the CADN bit provides a global clear function, forcing all DONE bits to be cleared.

If the NOP bit is set, the command is ignored. This allows you to set a single, byte-wide register with a 32-bit write while not affecting the other registers addressed in the write. In such a case the other three bytes of the word would all have their NOP bit set so that that these register will not be affected by the write.

Reads of this register return all zeroes.

25.3.5.10.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	0	0					0	
W	NOP	CADN	0				CDNE	
Reset	0	0	0	0	0	0	0	0

25.3.5.10.4 Fields

Field	Function
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register
6 CADN	Clears All DONE Bits 0b - Clears only the TCDn_CSR[DONE] bit specified in the CDNE field 1b - Clears all bits in TCDn_CSR[DONE]
5-2 —	Reserved
1-0 CDNE	Clear DONE Bit Clears the corresponding bit in TCDn_CSR[DONE]

25.3.5.11 Set START Bit Register (SSRT)

25.3.5.11.1 Offset

Register	Offset
SSRT	1Dh

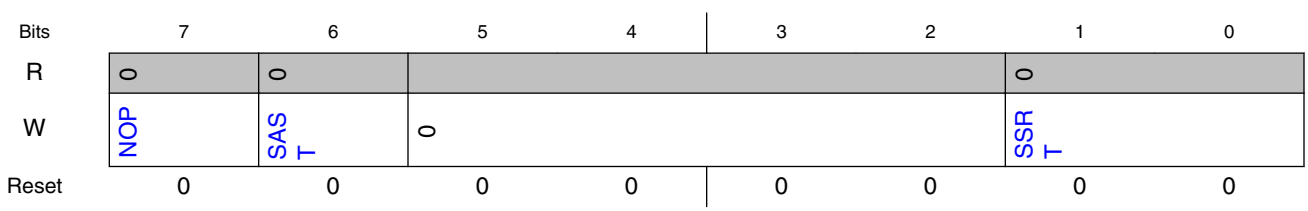
25.3.5.11.2 Function

The SSRT provides a simple memory-mapped mechanism to set the START bit in the TCD of the given channel. The data value on a register write causes the START bit in the corresponding transfer control descriptor to be set. Setting the SAST bit provides a global set function, forcing all START bits to be set.

If the NOP bit is set, the command is ignored. This allows you to set a single, byte-wide register with a 32-bit write while not affecting the other registers addressed in the write. In such a case the other three bytes of the word would all have their NOP bit set so that that these register will not be affected by the write.

Reads of this register return all zeroes.

25.3.5.11.3 Diagram



25.3.5.11.4 Fields

Field	Function
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register
6 SAST	Set All START Bits (activates all channels) 0b - Set only the TCDn_CSR[START] bit specified in the SSRT field 1b - Set all bits in TCDn_CSR[START]
5-2 —	Reserved
1-0 SSRT	Set START Bit Sets the corresponding bit in TCDn_CSR[START]

25.3.5.12 Clear Error Register (CERR)

25.3.5.12.1 Offset

Register	Offset
CERR	1Eh

25.3.5.12.2 Function

The CERR provides a simple memory-mapped mechanism to clear a given bit in the ERR to disable the error condition flag for a given channel. The given value on a register write causes the corresponding bit in the ERR to be cleared. Setting the CAEI bit provides a global clear function, forcing the ERR contents to be cleared, clearing all channel error indicators. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

25.3.5.12.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	0	0					0	
W	NOP	CAEI	0				CERR	
Reset	0	0	0	0	0	0	0	0

25.3.5.12.4 Fields

Field	Function
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register
6 CAEI	Clear All Error Indicators 0b - Clear only the ERR bit specified in the CERR field 1b - Clear all bits in ERR
5-2 —	Reserved
1-0 CERR	Clear Error Indicator Clears the corresponding bit in ERR

25.3.5.13 Clear Interrupt Request Register (CINT)

25.3.5.13.1 Offset

Register	Offset
CINT	1Fh

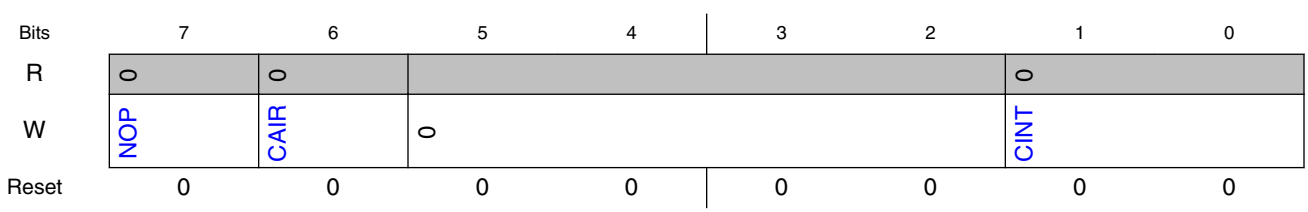
25.3.5.13.2 Function

The CINT provides a simple, memory-mapped mechanism to clear a given bit in the INT to disable the interrupt request for a given channel. The given value on a register write causes the corresponding bit in the INT to be cleared. Setting the CAIR bit provides a global clear function, forcing the entire contents of the INT to be cleared, disabling all DMA interrupt requests.

If the NOP bit is set, the command is ignored. This allows you to set a single, byte-wide register with a 32-bit write while not affecting the other registers addressed in the write. In such a case the other three bytes of the word would all have their NOP bit set so that that these register will not be affected by the write.

Reads of this register return all zeroes.

25.3.5.13.3 Diagram



25.3.5.13.4 Fields

Field	Function
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register
6 CAIR	Clear All Interrupt Requests 0b - Clear only the INT bit specified in the CINT field 1b - Clear all bits in INT

Table continues on the next page...

Field	Function
5-2 —	Reserved
1-0 CINT	Clear Interrupt Request Clears the corresponding bit in INT

25.3.5.14 Interrupt Request Register (INT)

25.3.5.14.1 Offset

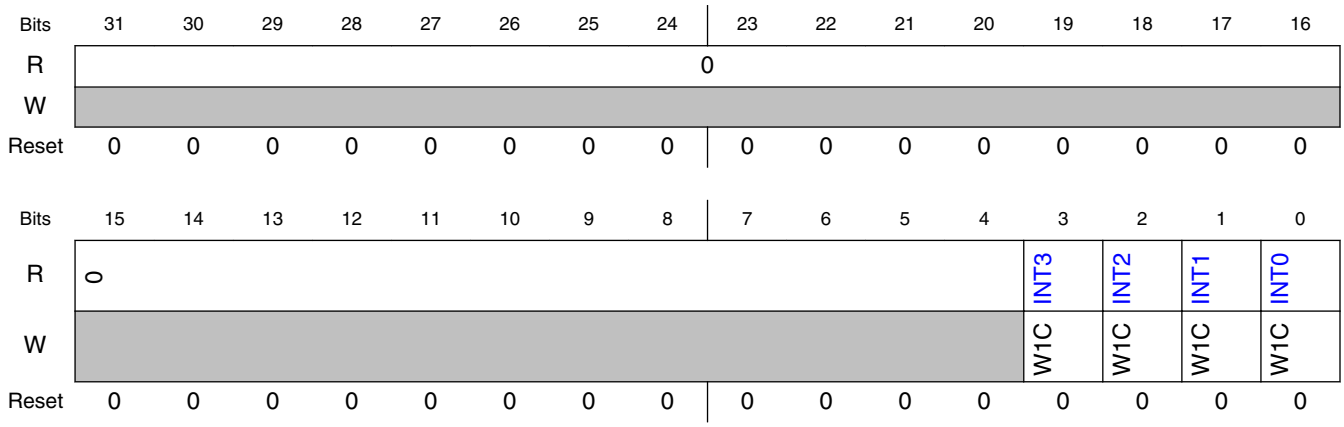
Register	Offset
INT	24h

25.3.5.14.2 Function

The INT register provides a bit map for the 4 channels signaling the presence of an interrupt request for each channel. Depending on the appropriate bit setting in the transfer-control descriptors, the eDMA engine generates an interrupt on data transfer completion. The outputs of this register are directly routed to the interrupt controller. During the interrupt-service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. Typically, a write to the CINT register in the interrupt service routine is used for this purpose.

The state of any given channel's interrupt request is directly affected by writes to this register; it is also affected by writes to the CINT register. On writes to INT, a 1 in any bit position clears the corresponding channel's interrupt request. A zero in any bit position has no affect on the corresponding channel's current interrupt status. The CINT register is provided so the interrupt request for a single channel can easily be cleared without the need to perform a read-modify-write sequence to the INT register.

25.3.5.14.3 Diagram



25.3.5.14.4 Fields

Field	Function
31-4 —	Reserved
3 INT3	Interrupt Request 3 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
2 INT2	Interrupt Request 2 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
1 INT1	Interrupt Request 1 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
0 INT0	Interrupt Request 0 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active

25.3.5.15 Error Register (ERR)

25.3.5.15.1 Offset

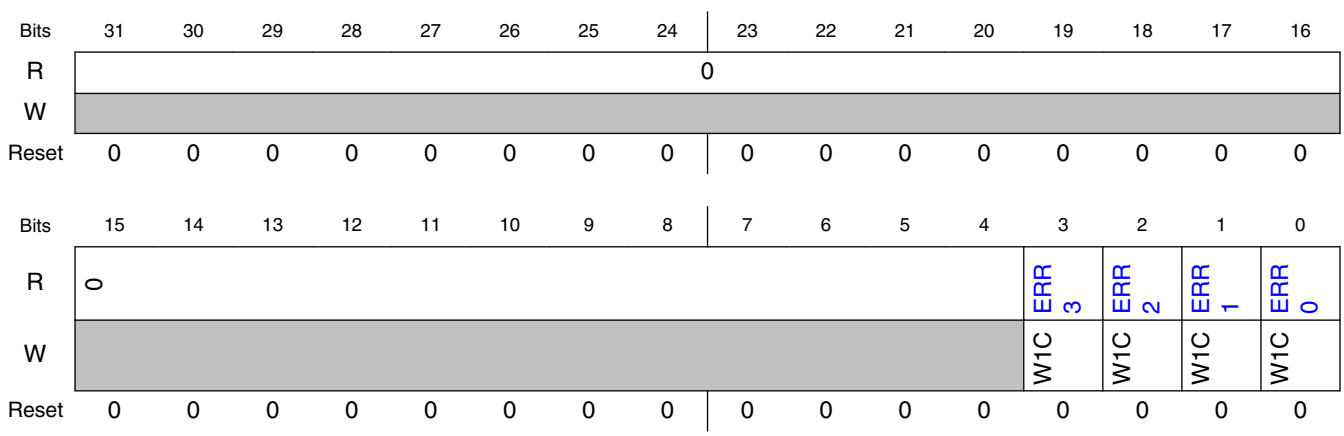
Register	Offset
ERR	2Ch

25.3.5.15.2 Function

The ERR register provides a bit map for the 4 channels, signaling the presence of an error for each channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate bit in this register. The outputs of this register are enabled by the contents of the EEI register, and then routed to the interrupt controller. During the execution of the interrupt-service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. Typically, a write to the CERR in the interrupt-service routine is used for this purpose. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when an error is detected.

The contents of this register can also be polled because a non-zero value indicates the presence of a channel error regardless of the state of the EEI fields. The state of any given channel's error indicators is affected by writes to this register; it is also affected by writes to the CERR. On writes to the ERR, a one in any bit position clears the corresponding channel's error status. A zero in any bit position has no affect on the corresponding channel's current error status. The CERR is provided so the error indicator for a single channel can easily be cleared.

25.3.5.15.3 Diagram



25.3.5.15.4 Fields

Field	Function
31-4 —	Reserved
3	Error In Channel 3 0b - An error in this channel has not occurred

Table continues on the next page...

Field	Function
ERR3	1b - An error in this channel has occurred
2 ERR2	Error In Channel 2 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
1 ERR1	Error In Channel 1 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
0 ERR0	Error In Channel 0 0b - An error in this channel has not occurred 1b - An error in this channel has occurred

25.3.5.16 Hardware Request Status Register (HRS)

25.3.5.16.1 Offset

Register	Offset
HRS	34h

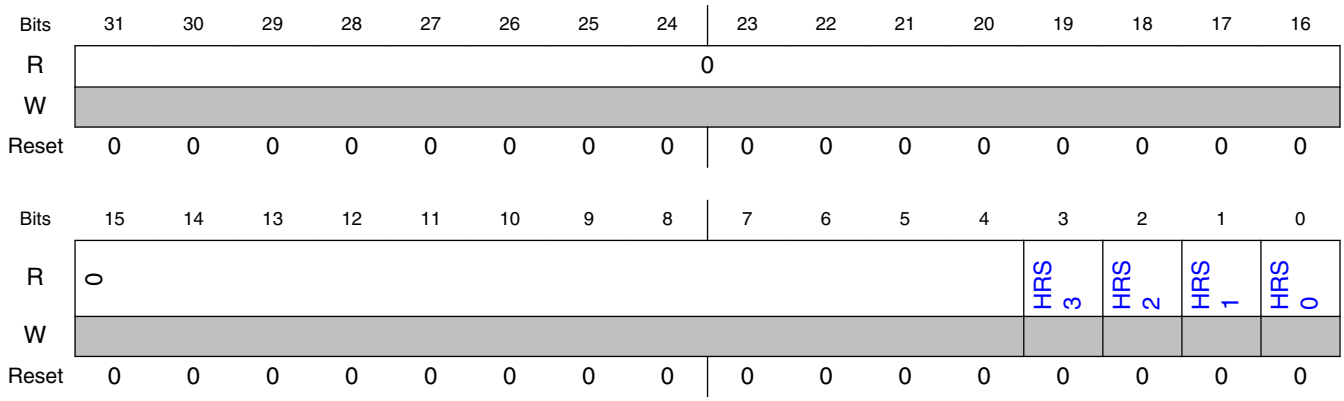
25.3.5.16.2 Function

The HRS register provides a bit map for the DMA channels, signaling the presence of a hardware request for each channel. The hardware request status bits reflect the current state of the register and qualified (via the ERQ fields) DMA request signals as seen by the DMA's arbitration logic. This view into the hardware request signals may be used for debug purposes.

NOTE

These bits reflect the state of the request as seen by the arbitration logic. Therefore, this status is affected by the ERQ bits.

25.3.5.16.3 Diagram



25.3.5.16.4 Fields

Field	Function
31-4 —	Reserved
3 HRS3	Hardware Request Status Channel 3 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0b - A hardware service request for channel 3 is not present 1b - A hardware service request for channel 3 is present
2 HRS2	Hardware Request Status Channel 2 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0b - A hardware service request for channel 2 is not present 1b - A hardware service request for channel 2 is present
1 HRS1	Hardware Request Status Channel 1 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0b - A hardware service request for channel 1 is not present 1b - A hardware service request for channel 1 is present
0 HRS0	Hardware Request Status Channel 0 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0b - A hardware service request for channel 0 is not present 1b - A hardware service request for channel 0 is present

25.3.5.17 Enable Asynchronous Request in Stop Register (EARS)

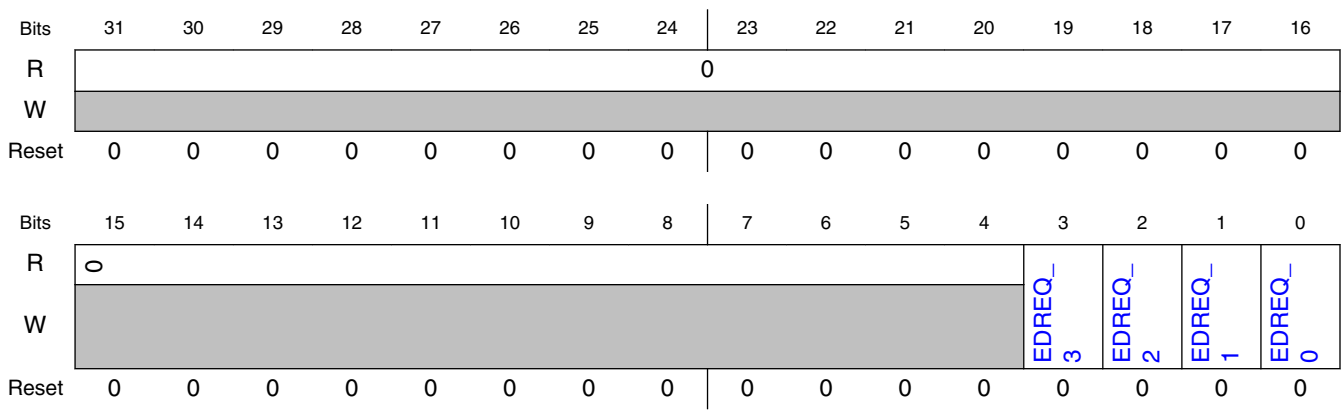
25.3.5.17.1 Offset

Register	Offset
EARS	44h

25.3.5.17.2 Function

The EARS register is used to enable or disable the DMA requests in Enable Request Register (ERQ) by AND'ing the bits of these two registers.

25.3.5.17.3 Diagram



25.3.5.17.4 Fields

Field	Function
31-4 —	Reserved
3 EDREQ_3	Enable asynchronous DMA request in stop mode for channel 3. 0b - Disable asynchronous DMA request for channel 3. 1b - Enable asynchronous DMA request for channel 3.
2 EDREQ_2	Enable asynchronous DMA request in stop mode for channel 2. 0b - Disable asynchronous DMA request for channel 2. 1b - Enable asynchronous DMA request for channel 2.
1 EDREQ_1	Enable asynchronous DMA request in stop mode for channel 1. 0b - Disable asynchronous DMA request for channel 1 1b - Enable asynchronous DMA request for channel 1.
0 EDREQ_0	Enable asynchronous DMA request in stop mode for channel 0. 0b - Disable asynchronous DMA request for channel 0. 1b - Enable asynchronous DMA request for channel 0.

25.3.5.18 Channel Priority Register (DCHPRI0 - DCHPRI3)

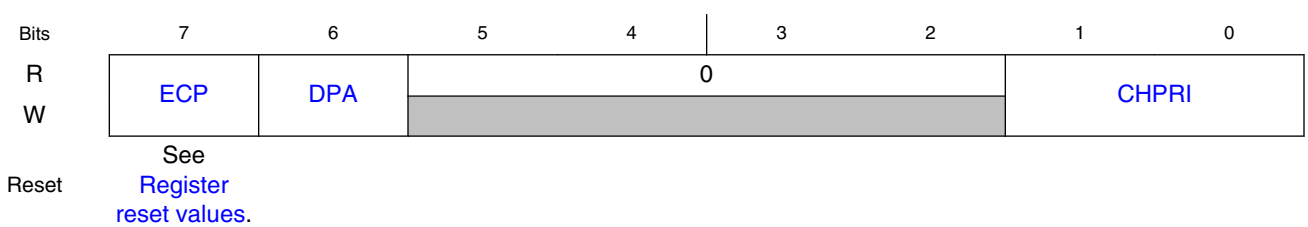
25.3.5.18.1 Offset

Register	Offset
DCHPRI3	100h
DCHPRI2	101h
DCHPRI1	102h
DCHPRI0	103h

25.3.5.18.2 Function

When fixed-priority channel arbitration is enabled (CR[ERCA] = 0), the contents of these registers define the unique priorities associated with each channel. The channel priorities are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, etc. Software must program the channel priorities with unique values; otherwise, a configuration error is reported. The range of the priority value is limited to the values of 0 through 3.

25.3.5.18.3 Diagram



25.3.5.18.4 Register reset values

Register	Reset value
DCHPRI0	00h
DCHPRI1	01h
DCHPRI2	02h
DCHPRI3	03h

25.3.5.18.5 Fields

Field	Function
7 ECP	Enable Channel Preemption. This field resets to 0. 0b - Channel n cannot be suspended by a higher priority channel's service request. 1b - Channel n can be temporarily suspended by the service request of a higher priority channel.
6 DPA	Disable Preempt Ability. This field resets to 0. 0b - Channel n can suspend a lower priority channel. 1b - Channel n cannot suspend any channel, regardless of channel priority.
5-2 —	Reserved
1-0 CHPRI	Channel n Arbitration Priority Channel priority when fixed-priority arbitration is enabled

25.3.5.19 TCD Source Address (TCD0_SADDR - TCD3_SADDR)

25.3.5.19.1 Offset

Register	Offset
TCD0_SADDR	1000h
TCD1_SADDR	1020h
TCD2_SADDR	1040h
TCD3_SADDR	1060h

25.3.5.19.2 Function

This register contains the source address of the transfer.

25.3.5.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

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

25.3.5.19.4 Fields

Field	Function
31-0 SADDR	Source Address Memory address pointing to the source data.

25.3.5.20 TCD Signed Source Address Offset (TCD0_SOFF - TCD3_SOFF)

25.3.5.20.1 Offset

Register	Offset
TCD0_SOFF	1004h
TCD1_SOFF	1024h
TCD2_SOFF	1044h
TCD3_SOFF	1064h

25.3.5.20.2 Diagram

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

25.3.5.20.3 Fields

Field	Function
15-0 SOFF	Source address signed offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

25.3.5.21 TCD Transfer Attributes (TCD0_ATTR - TCD3_ATTR)

25.3.5.21.1 Offset

Register	Offset
TCD0_ATTR	1006h
TCD1_ATTR	1026h
TCD2_ATTR	1046h
TCD3_ATTR	1066h

25.3.5.21.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SMOD					SSIZE			DMOD					DSIZE		
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

25.3.5.21.3 Fields

Field	Function
15-11 SMOD	Source Address Modulo 00000b - Source address modulo feature is disabled 00001-11111b - This value defines a specific address range specified to be the value after SADDR + SOFF calculation is performed on the original register value. Setting this field provides the ability to implement a circular data queue easily. For data queues requiring power-of-2 size bytes, the queue should start at a 0-modulo-size address and the SMOD field should be set to the appropriate value for the queue, freezing the desired number of upper address bits. The value programmed into this field specifies the number of lower address bits allowed to change. For a circular queue application, the SOFF is typically set to the transfer size to implement post-increment addressing with the SMOD function constraining the addresses to a 0-modulo-size range.
10-8 SSIZE	Source data transfer size **NOTE:** Using a Reserved value causes a configuration error. **NOTE:** The eDMA defaults to privileged data access for all transactions. 000b - 8-bit 001b - 16-bit 010b - 32-bit 011b - Reserved 100b - 16-byte 101b - 32-byte 110b - Reserved 111b - Reserved
7-3 DMOD	Destination Address Modulo See the SMOD definition
2-0	Destination data transfer size

Field	Function
DSIZE	See the SSIZE definition

25.3.5.22 TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD0_NBYTES_MLNO - TCD3_NBYTES_MLNO)

25.3.5.22.1 Offset

Register	Offset
TCD0_NBYTES_MLNO	1008h
TCD1_NBYTES_MLNO	1028h
TCD2_NBYTES_MLNO	1048h
TCD3_NBYTES_MLNO	1068h

25.3.5.22.2 Function

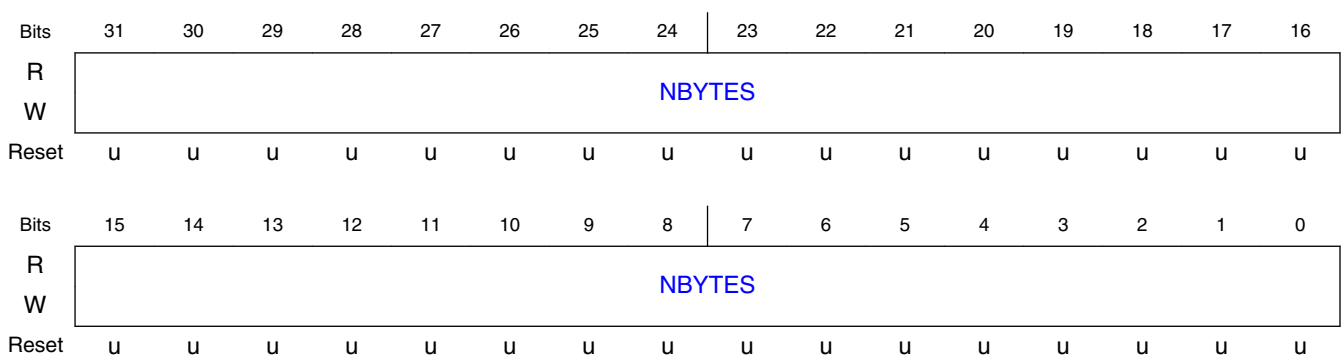
This register, or one of the next two registers (TCD_NBYTES_MLOFFNO, TCD_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is disabled (CR[EMLM] = 0)

If minor loop mapping is enabled, see the TCD_NBYTES_MLOFFNO and TCD_NBYTES_MLOFFYES register descriptions for the definition of TCD word 2.

25.3.5.22.3 Diagram



25.3.5.22.4 Fields

Field	Function
31-0 NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed. **NOTE:** An NBYTES value of 0x0000_0000 is interpreted as a 4 GB transfer.

25.3.5.23 TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD0_NBYTES_MLOFFNO - TCD3_NBYTES_MLOFFNO)

25.3.5.23.1 Offset

Register	Offset
TCD0_NBYTES_MLOFFNO	1008h
TCD1_NBYTES_MLOFFNO	1028h
TCD2_NBYTES_MLOFFNO	1048h
TCD3_NBYTES_MLOFFNO	1068h

25.3.5.23.2 Function

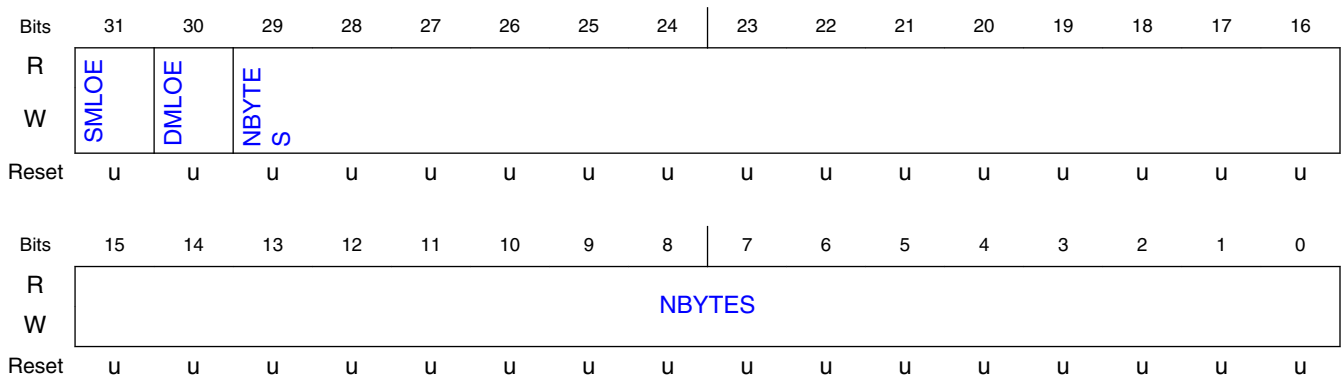
One of three registers (this register, TCD_NBYTES_MLNO, or TCD_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- SMLOE = 0 and DMLOE = 0

If minor loop mapping is enabled and SMLOE or DMLOE is set, then refer to the TCD_NBYTES_MLOFFYES register description. If minor loop mapping is disabled, then refer to the TCD_NBYTES_MLNO register description.

25.3.5.23.3 Diagram



25.3.5.23.4 Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - The minor loop offset is not applied to the SADDR 1b - The minor loop offset is applied to the SADDR
30 DMLOE	Destination Minor Loop Offset enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - The minor loop offset is not applied to the DADDR 1b - The minor loop offset is applied to the DADDR
29-0 NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

25.3.5.24 TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD0_NBYTES_MLOFFYES - TCD3_NBYTES_MLOFFYES)

25.3.5.24.1 Offset

Register	Offset
TCD0_NBYTES_MLOFFYES	1008h
TCD1_NBYTES_MLOFFYES	1028h
TCD2_NBYTES_MLOFFYES	1048h
TCD3_NBYTES_MLOFFYES	1068h

25.3.5.24.2 Function

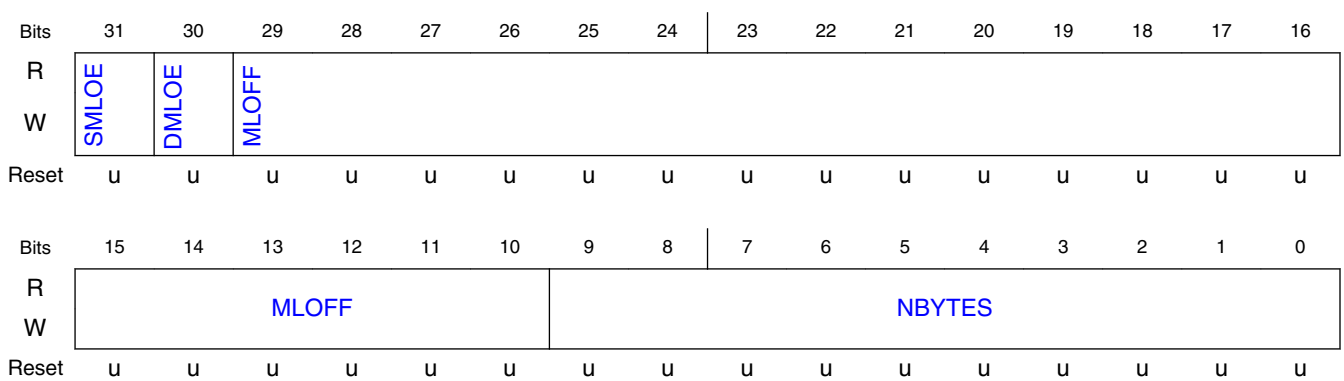
One of three registers (this register, TCD_NBYTES_MLNO, or TCD_NBYTES_MLOFFNO), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- Minor loop offset is enabled (SMLOE or DMLOE = 1)

If minor loop mapping is enabled and SMLOE and DMLOE are cleared, then refer to the TCD_NBYTES_MLOFFNO register description. If minor loop mapping is disabled, then refer to the TCD_NBYTES_MLNO register description.

25.3.5.24.3 Diagram



25.3.5.24.4 Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - The minor loop offset is not applied to the SADDR 1b - The minor loop offset is applied to the SADDR
30 DMLOE	Destination Minor Loop Offset enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - The minor loop offset is not applied to the DADDR 1b - The minor loop offset is applied to the DADDR
29-10 MLOFF	If SMLOE or DMLOE is set, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9-0 NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

25.3.5.25 TCD Last Source Address Adjustment (TCD0_SLAST - TCD3_SLAST)

25.3.5.25.1 Offset

Register	Offset
TCD0_SLAST	100Ch
TCD1_SLAST	102Ch
TCD2_SLAST	104Ch
TCD3_SLAST	106Ch

25.3.5.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SLAST															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

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

25.3.5.25.3 Fields

Field	Function
31-0 SLAST	Last Source Address Adjustment Adjustment value added to the source address at the completion of the major iteration count. This value can be applied to restore the source address to the initial value, or adjust the address to reference the next data structure. This register uses two's complement notation; the overflow bit is discarded.

25.3.5.26 TCD Destination Address (TCD0_DADDR - TCD3_DADDR)

25.3.5.26.1 Offset

Register	Offset
TCD0_DADDR	1010h
TCD1_DADDR	1030h
TCD2_DADDR	1050h
TCD3_DADDR	1070h

25.3.5.26.2 Function

This register contains the destination address of the transfer.

25.3.5.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

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

25.3.5.26.4 Fields

Field	Function
31-0 DADDR	Destination Address Memory address pointing to the destination data.

25.3.5.27 TCD Signed Destination Address Offset (TCD0_DOFF - TCD3_DOFF)

25.3.5.27.1 Offset

Register	Offset
TCD0_DOFF	1014h
TCD1_DOFF	1034h
TCD2_DOFF	1054h
TCD3_DOFF	1074h

25.3.5.27.2 Diagram

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

25.3.5.27.3 Fields

Field	Function
15-0 DOFF	Destination Address Signed Offset Sign-extended offset applied to the current destination address to form the next-state value as each destination write is completed.

25.3.5.28 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD3_CITER_ELINKNO)

25.3.5.28.1 Offset

Register	Offset
TCD0_CITER_ELINKNO	1016h
TCD1_CITER_ELINKNO	1036h
TCD2_CITER_ELINKNO	1056h
TCD3_CITER_ELINKNO	1076h

25.3.5.28.2 Function

This register contains the minor-loop channel-linking configuration and the channel's current iteration count. It is the same register as TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD3_CITER_ELINKYES), but its fields are defined differently based on the state of the ELINK field. If the ELINK field is cleared, this register is defined as follows.

25.3.5.28.3 Diagram

25.3.5.28.4 Fields

Field	Function
15 ELINK	Enable channel-to-channel linking on minor-loop complete As the channel completes the minor loop, this flag enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** This bit must be equal to the BITER[ELINK] bit; otherwise, a configuration error is reported. 0b - The channel-to-channel linking is disabled 1b - The channel-to-channel linking is enabled
14-0 CITER	Current Major Iteration Count This field is the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations, for example, final source and destination address calculations, optionally generating an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE:** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE:** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

25.3.5.29 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD3_CITER_ELINKYES)

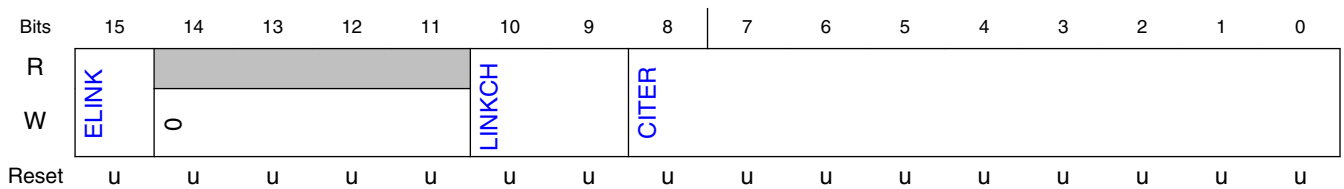
25.3.5.29.1 Offset

Register	Offset
TCD0_CITER_ELINKYES	1016h
TCD1_CITER_ELINKYES	1036h
TCD2_CITER_ELINKYES	1056h
TCD3_CITER_ELINKYES	1076h

25.3.5.29.2 Function

This register contains the minor-loop channel-linking configuration and the channel's current iteration count. It is the same register as TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD3_CITER_ELINKNO), but its fields are defined differently based on the state of the ELINK field. If the ELINK field is set, this register is defined as follows.

25.3.5.29.3 Diagram



25.3.5.29.4 Fields

Field	Function
15 ELINK	Enable channel-to-channel linking on minor-loop complete As the channel completes the minor loop, this flag enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** This bit must be equal to the BITER[ELINK] bit; otherwise, a configuration error is reported. 0b - The channel-to-channel linking is disabled 1b - The channel-to-channel linking is enabled
14-11 —	Reserved
10-9 LINKCH	Minor Loop Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request to the channel defined by this field by setting that channel's TCDn_CSR[START] bit.
8-0 CITER	Current Major Iteration Count This field is the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations, for example, final source and destination address calculations, optionally generating an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE:** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE:** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

25.3.5.30 TCD Last Destination Address Adjustment/Scatter Gather Address (TCD0_DLASTSGA - TCD3_DLASTSGA)

25.3.5.30.1 Offset

Register	Offset
TCD0_DLASTSGA	1018h
TCD1_DLASTSGA	1038h
TCD2_DLASTSGA	1058h
TCD3_DLASTSGA	1078h

25.3.5.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLASTSGA															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

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

25.3.5.30.3 Fields

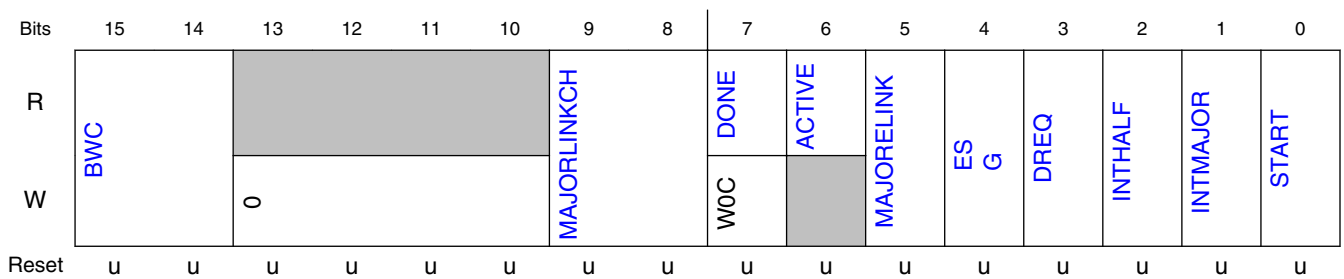
Field	Function
31-0 DLASTSGA	DLASTSGA Destination last address adjustment or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather). If (TCDn_CSR[ESG] = 0) then: • Adjustment value added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo-32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo-32-byte, otherwise a configuration error is reported.

25.3.5.31 TCD Control and Status (TCD0_CSR - TCD3_CSR)

25.3.5.31.1 Offset

Register	Offset
TCD0_CSR	101Ch
TCD1_CSR	103Ch
TCD2_CSR	105Ch
TCD3_CSR	107Ch

25.3.5.31.2 Diagram



25.3.5.31.3 Fields

Field	Function
15-14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces the eDMA to stall after the completion of each read/write access to control the bus request bandwidth seen by the crossbar switch. **NOTE:** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. **NOTE:** When executing a large, zero wait-stated memory-to-memory transfer, insert bandwidth control using the TCD_CSR[BWC] bits to avoid: • Starvation of another master accessing the memory. • Any delay in writing a TCD during the transfer. 00b - No eDMA engine stalls. 01b - Reserved 10b - eDMA engine stalls for 4 cycles after each R/W. 11b - eDMA engine stalls for 8 cycles after each R/W.
13-10	Reserved

Table continues on the next page...

Field	Function
—	
9-8 MAJORLINKCH	Major Loop Link Channel Number If (MAJORELINK = 0) then: • No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted. Otherwise: • After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] bit.
7 DONE	Channel Done This flag indicates the eDMA has completed the major loop. The eDMA engine sets it as the CITER count reaches zero. The software clears it, or the hardware when the channel is activated. **NOTE:** This bit must be cleared to write the MAJORELINK or ESG bits.
6 ACTIVE	Channel Active This flag signals the channel is currently in execution. It is set when channel service begins, and is cleared by the eDMA as the minor loop completes or when any error condition is detected.
5 MAJORELINK	Enable channel-to-channel linking on major loop complete As the channel completes the major loop, this flag enables the linking to another channel, defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. **NOTE:** To support the dynamic linking coherency model, this field is forced to zero when written to while the TCDn_CSR[DONE] bit is set. 0b - The channel-to-channel linking is disabled. 1b - The channel-to-channel linking is enabled.
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses DLASTSGA as a memory pointer to a 0-modulo-32 address containing a 32-byte data structure loaded as the transfer control descriptor into the local memory. **NOTE:** To support the dynamic scatter/gather coherency model, this field is forced to zero when written to while the TCDn_CSR[DONE] bit is set. 0b - The current channel's TCD is normal format. 1b - The current channel's TCD specifies a scatter gather format. The DLASTSGA field provides a memory pointer to the next TCD to be loaded into this channel after the major loop completes its execution.
3 DREQ	Disable Request If this flag is set, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches zero. 0b - The channel's ERQ bit is not affected. 1b - The channel's ERQ bit is cleared when the major loop is complete.
2 INTHALF	Enable an interrupt when major counter is half complete. If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the INT register when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER == (BITER >> 1)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE:** If BITER = 1, do not use INTHALF. Use INTMAJOR instead. 0b - The half-point interrupt is disabled. 1b - The half-point interrupt is enabled.
1	Enable an interrupt when major iteration count completes.

Table continues on the next page...

Field	Function
INTMAJOR	If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the INT when the current major iteration count reaches zero. 0b - The end-of-major loop interrupt is disabled. 1b - The end-of-major loop interrupt is enabled.
0 START	Channel Start If this flag is set, the channel is requesting service. The eDMA hardware automatically clears this flag after the channel begins execution. 0b - The channel is not explicitly started. 1b - The channel is explicitly started via a software initiated service request.

25.3.5.32 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD3_BITER_ELINKNO)

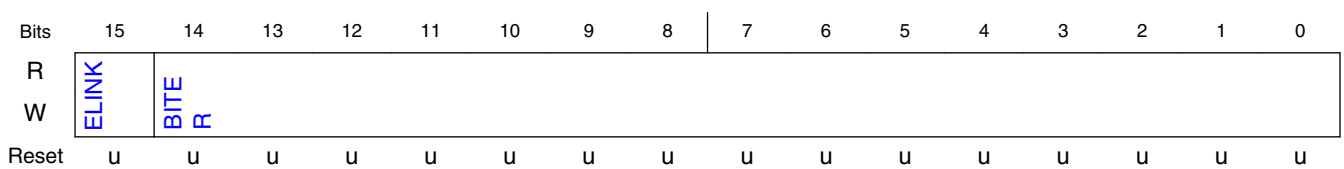
25.3.5.32.1 Offset

Register	Offset
TCD0_BITER_ELINKNO	101Eh
TCD1_BITER_ELINKNO	103Eh
TCD2_BITER_ELINKNO	105Eh
TCD3_BITER_ELINKNO	107Eh

25.3.5.32.2 Function

If the TCDn_BITER[ELINK] bit is cleared, the TCDn_BITER register is defined as follows.

25.3.5.32.3 Diagram



25.3.5.32.4 Fields

Field	Function
15	Enables channel-to-channel linking on minor loop complete

Table continues on the next page...

Field	Function
ELINK	As the channel completes the minor loop, this flag enables the linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. 0b - The channel-to-channel linking is disabled 1b - The channel-to-channel linking is enabled
14-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

25.3.5.33 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD3 _BITER_ELINKYES)

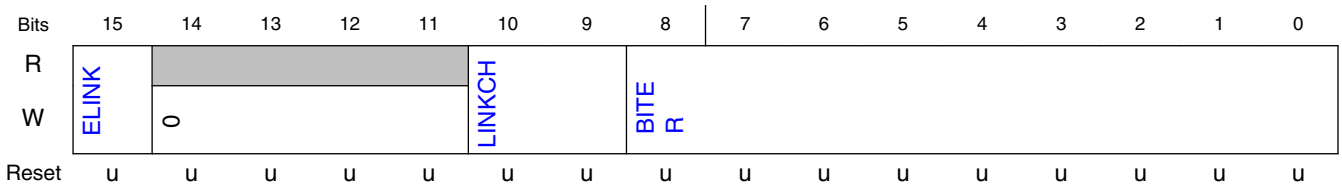
25.3.5.33.1 Offset

Register	Offset
TCD0_BITER_ELINK YES	101Eh
TCD1_BITER_ELINK YES	103Eh
TCD2_BITER_ELINK YES	105Eh
TCD3_BITER_ELINK YES	107Eh

25.3.5.33.2 Function

If the TCDn_BITER[ELINK] bit is set, the TCDn_BITER register is defined as follows.

25.3.5.33.3 Diagram



25.3.5.33.4 Fields

Field	Function
15 ELINK	Enables channel-to-channel linking on minor loop complete As the channel completes the minor loop, this flag enables the linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. 0b - The channel-to-channel linking is disabled 1b - The channel-to-channel linking is enabled
14-11 —	Reserved
10-9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] bit. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field.
8-0 BITER	Starting major iteration count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

25.4 Functional description

The operation of the eDMA is described in the following subsections.

25.4.1 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in the following diagram, the first segment involves the channel activation:

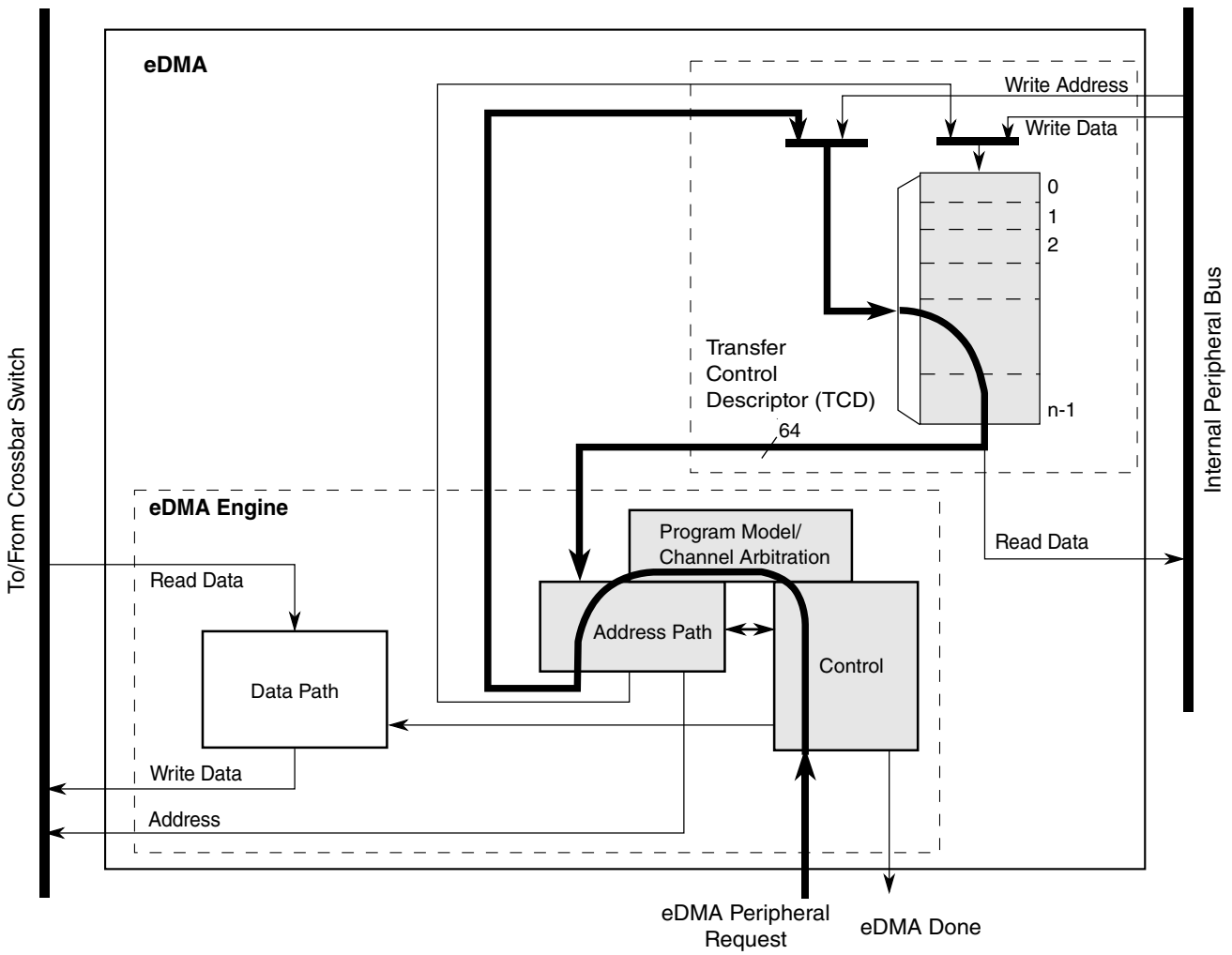


Figure 25-2. eDMA operation, part 1

This example uses the assertion of the eDMA peripheral request signal to request service for channel *n*. Channel activation via software and the TCD*n*_CSR[START] bit follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, then into the program model and channel arbitration. In the next cycle, the channel arbitration performs, using the fixed-priority or round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the local memory for TCD*n*. Next, the TCD memory is accessed and the required descriptor read from the local memory and loaded into the

eDMA engine address path channel x or y registers. The TCD memory is 64 bits wide to minimize the time needed to fetch the activated channel descriptor and load it into the address path channel x or y registers.

The following diagram illustrates the second part of the basic data flow:

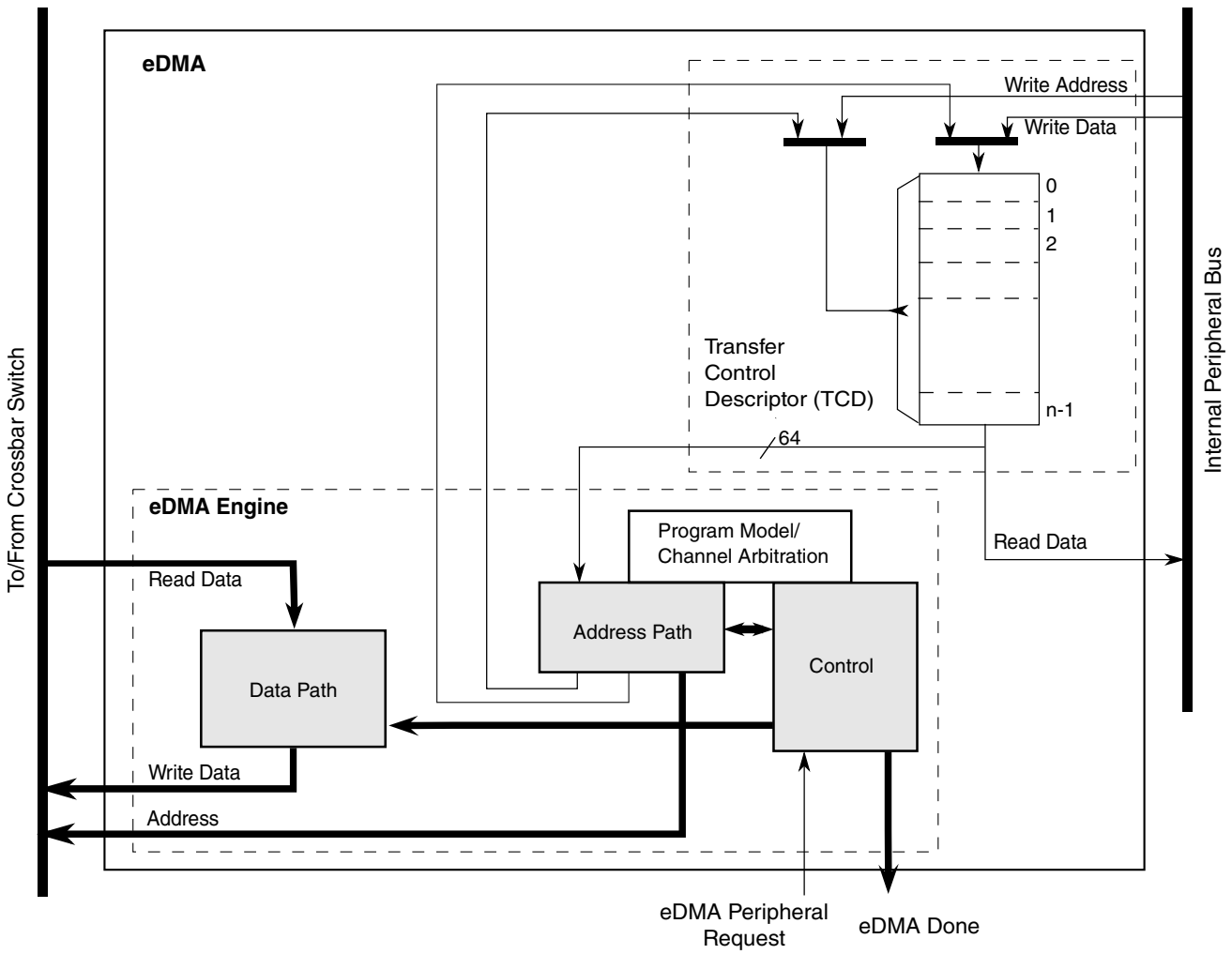


Figure 25-3. eDMA operation, part 2

The modules associated with the data transfer (address path, data path, and control) sequence through the required source reads and destination writes to perform the actual data movement. The source reads are initiated and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read/destination write processing continues until the minor byte count has transferred.

After the minor byte count has moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to certain fields in the appropriate TCD, for example, SADDR, DADDR, CITER. If the major iteration count is exhausted, additional operations are performed. These include the final address

adjustments and reloading of the BITER field into the CITER. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). The updates to the TCD memory and the assertion of an interrupt request are shown in the following diagram.

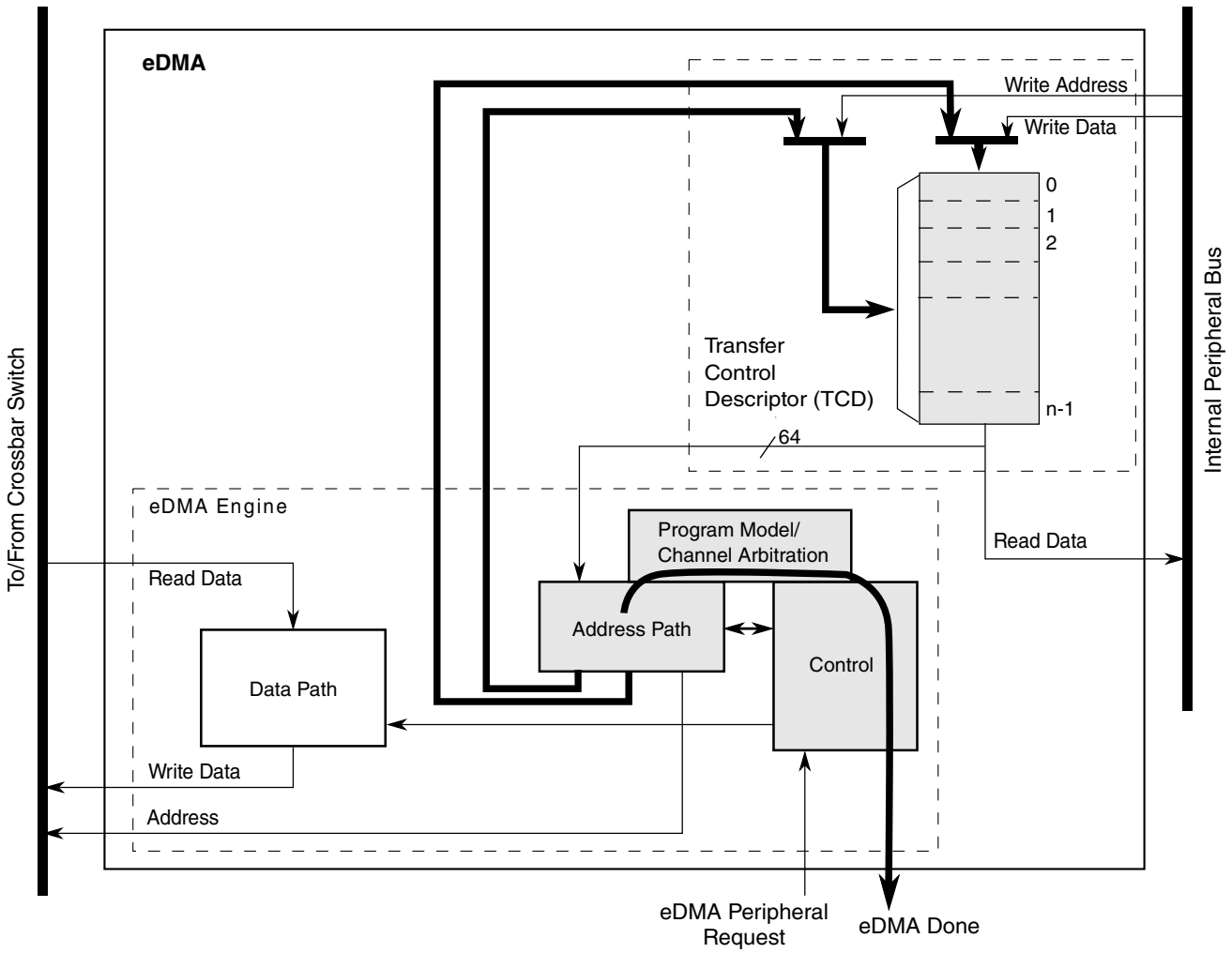


Figure 25-4. eDMA operation, part 3

25.4.2 Fault reporting and handling

Channel errors are reported in the Error Status register (DMAx_ES) and can be caused by:

- A configuration error, which is an illegal setting in the transfer-control descriptor or an illegal priority register setting in Fixed-Arbitration mode, or
- An error termination to a bus master read or write cycle

A configuration error is reported when the starting source or destination address, source or destination offsets, minor loop byte count, or the transfer size represent an inconsistent state. Each of these possible causes are detailed below:

- The addresses and offsets must be aligned on 0-modulo-transfer-size boundaries.
- The minor loop byte count must be a multiple of the source and destination transfer sizes.
- All source reads and destination writes must be configured to the natural boundary of the programmed transfer size respectively.
- In fixed arbitration mode, a configuration error is caused by any two channel priorities being equal. All channel priority levels must be unique when fixed arbitration mode is enabled.

NOTE

When two channels have the same priority, a channel priority error exists and will be reported in the Error Status register. However, the channel number will not be reported in the Error Status register. When all of the channel priorities within a group are not unique, the channel number selected by arbitration in undetermined.

To aid in Channel Priority Error (CPE) debug, set the Halt On Error bit in the DMA's Control Register. If all of the channel priorities within a group are not unique, the DMA will be halted after the CPE error is recorded. The DMA will remain halted and will not process any channel service requests. Once all of the channel priorities are set to unique numbers, the DMA may be enabled again by clearing the Halt bit.

- If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (DLAST_SGA) is not aligned on a 32-byte boundary.
- If minor loop channel linking is enabled upon channel completion, a configuration error is reported when the link is attempted if the TCDn_CITER[E_LINK] bit does not equal the TCDn_BITER[E_LINK] bit.

If enabled, all configuration error conditions, except the scatter/gather and minor-loop link errors, report as the channel activates and asserts an error interrupt request. A scatter/gather configuration error is reported when the scatter/gather operation begins at major loop completion when properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

If a system bus read or write is terminated with an error, the data transfer is stopped and the appropriate bus error flag set. In this case, the state of the channel's transfer control descriptor is updated by the eDMA engine with the current source address, destination address, and current iteration count at the point of the fault. When a system bus error occurs, the channel terminates after the next transfer. Due to pipeline effect, the next transfer is already in progress when the bus error is received by the eDMA. If a bus error occurs on the last read prior to beginning the write sequence, the write executes using the data captured during the bus error. If a bus error occurs on the last write prior to switching to the next read sequence, the read sequence executes before the channel terminates due to the destination bus error.

A transfer may be cancelled by software with the CR[CX] bit. When a cancel transfer request is recognized, the DMA engine stops processing the channel. The current read-write sequence is allowed to finish. If the cancel occurs on the last read-write sequence of a major or minor loop, the cancel request is discarded and the channel retires normally.

The error cancel transfer is the same as a cancel transfer except the Error Status register (DMAx_ES) is updated with the cancelled channel number and ECX is set. The TCD of a cancelled channel contains the source and destination addresses of the last transfer saved in the TCD. If the channel needs to be restarted, you must re-initialize the TCD because the aforementioned fields no longer represent the original parameters. When a transfer is cancelled by the error cancel transfer mechanism, the channel number is loaded into DMA_ES[ERRCHN] and ECX and VLD are set. In addition, an error interrupt may be generated if enabled.

NOTE

The cancel transfer request allows the user to stop a large data transfer in the event the full data transfer is no longer needed. The cancel transfer bit does not abort the channel. It simply stops the transferring of data and then retires the channel through its normal shutdown sequence. The application software must handle the context of the cancel. If an interrupt is desired (or not), then the interrupt should be enabled (or disabled) before the cancel request. The application software must clean up the transfer control descriptor since the full transfer did not occur.

The occurrence of any error causes the eDMA engine to stop normal processing of the active channel immediately (it goes to its error processing states and the transaction to the system bus still has pipeline effect), and the appropriate channel bit in the eDMA error register is asserted. At the same time, the details of the error condition are loaded into the Error Status register (DMAx_ES). The major loop complete indicators, setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request,

are not affected when an error is detected. After the error status has been updated, the eDMA engine continues operating by servicing the next appropriate channel. A channel that experiences an error condition is not automatically disabled. If a channel is terminated by an error and then issues another service request before the error is fixed, that channel executes and terminates with the same error condition.

25.4.3 Channel preemption

Channel preemption is enabled on a per-channel basis by setting the DCHPRIn[ECP] bit. Channel preemption allows the executing channel's data transfers to temporarily suspend in favor of starting a higher priority channel. After the preempting channel has completed all its minor loop data transfers, the preempted channel is restored and resumes execution. After the restored channel completes one read/write sequence, it is again eligible for preemption. If any higher priority channel is requesting service, the restored channel is suspended and the higher priority channel is serviced. Nested preemption, that is, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot be preempted. Preemption is available only when fixed arbitration is selected.

A channel's ability to preempt another channel can be disabled by setting DCHPRIn[DPA]. When a channel's preempt ability is disabled, that channel cannot suspend a lower priority channel's data transfer, regardless of the lower priority channel's ECP setting. This allows for a pool of low priority, large data-moving channels to be defined. These low priority channels can be configured to not preempt each other, thus preventing a low priority channel from consuming the preempt slot normally available to a true, high priority channel.

25.4.4 Performance

This section addresses the performance of the eDMA module, focusing on two separate metrics:

- In the traditional data movement context, performance is best expressed as the peak data transfer rates achieved using the eDMA. In most implementations, this transfer rate is limited by the speed of the source and destination address spaces.
- In a second context where device-paced movement of single data values to/from peripherals is dominant, a measure of the requests that can be serviced in a fixed time is a more relevant metric. In this environment, the speed of the source and destination address spaces remains important. However, the microarchitecture of the eDMA also factors significantly into the resulting metric.

25.4.4.1 Peak transfer rates

The peak transfer rates for several different source and destination transfers are shown in the following tables. These tables assume:

- Internal SRAM can be accessed with zero wait-states when viewed from the system bus data phase
- All internal peripheral bus reads require two wait-states, and internal peripheral bus writes three wait-states, when viewed from the system bus data phase
- All internal peripheral bus accesses are 32-bits in size

NOTE

All architectures will not meet the assumptions listed above. See the SRAM configuration section for more information.

This table compares peak transfer rates based on different possible system speeds. Specific chips/devices may not support all system speeds listed.

Table 25-4. eDMA peak transfer rates (Mbytes/sec)

System Speed, Width	Internal SRAM-to-Internal SRAM	32 bit internal peripheral bus-to-Internal SRAM	Internal SRAM-to-32 bit internal peripheral bus
48.0 MHz, 32 bit	96.0	48.0	38.4
66.7 MHz, 32 bit	133.3	66.7	53.3
83.3 MHz, 32 bit	166.7	83.3	66.7
100.0 MHz, 32 bit	200.0	100.0	80.0
133.3 MHz, 32 bit	266.7	133.3	106.7
150.0 MHz, 32 bit	300.0	150.0	120.0

Internal-SRAM-to-internal-SRAM transfers occur at the core's datapath width. For all transfers involving the internal peripheral bus, 32-bit transfer sizes are used. In all cases, the transfer rate includes the time to read the source plus the time to write the destination.

25.4.4.2 Peak request rates

The second performance metric is a measure of the number of DMA requests that can be serviced in a given amount of time. For this metric, assume that the peripheral request causes the channel to move a single internal peripheral bus-mapped operand to/from

internal SRAM. The same timing assumptions used in the previous example apply to this calculation. In particular, this metric also reflects the time required to activate the channel.

The eDMA design supports the following hardware service request sequence. Note that the exact timing from Cycle 7 is a function of the response times for the channel's read and write accesses. In the case of an internal peripheral bus read and internal SRAM write, the combined data phase time is 4 cycles. For an SRAM read and internal peripheral bus write, it is 5 cycles.

Table 25-5. Hardware service request process

Cycle		Description
With internal peripheral bus read and internal SRAM write	With SRAM read and internal peripheral bus write	
1		eDMA peripheral request is asserted.
2		The eDMA peripheral request is registered locally in the eDMA module and qualified. TCD*n*_CSR[START] bit initiated requests start at this point with the registering of the user write to TCD*n* word 7.
3		Channel arbitration begins.
4		Channel arbitration completes. The transfer control descriptor local memory read is initiated.
5–6		The first two parts of the activated channel's TCD is read from the local memory. The memory width to the eDMA engine is 64 bits, so the entire descriptor can be accessed in four cycles
7		The first system bus read cycle is initiated, as the third part of the channel's TCD is read from the local memory. Depending on the state of the crossbar switch, arbitration at the system bus may insert an additional cycle of delay here.
8–11	8–12	The last part of the TCD is read in. This cycle represents the first data phase for the read, and the address phase for the destination write.
12	13	This cycle represents the data phase of the last destination write.
13	14	The eDMA engine completes the execution of the inner minor loop and prepares to write back the required TCD*n* fields into the local memory. The TCD*n* word 7 is read and checked for channel linking or scatter/gather requests.
14	15	The appropriate fields in the first part of the TCD*n* are written back into the local memory.
15	16	The fields in the second part of the TCD*n* are written back into the local memory. This cycle coincides with the next channel arbitration cycle start.
16	17	The next channel to be activated performs the read of the first part of its TCD from the local memory. This is equivalent to Cycle 4 for the first channel's service request.

Assuming zero wait states on the system bus, DMA requests can be processed every 9 cycles. Assuming an average of the access times associated with internal peripheral bus-to-SRAM (4 cycles) and SRAM-to-internal peripheral bus (5 cycles), DMA requests can be processed every 11.5 cycles (4 + (4+5)/2 + 3). This is the time from Cycle 4 to Cycle x +5. The resulting peak request rate, as a function of the system frequency, is shown in the following table.

Table 25-6. eDMA peak request rate (MReq/sec)

System frequency (MHz)	Request rate with zero wait states	Request rate with wait states
48.0	5.3	4.2
66.6	7.4	5.8
83.3	9.2	7.2
100.0	11.1	8.7
133.3	14.8	11.6
150.0	16.6	13.0

A general formula to compute the peak request rate with overlapping requests is:

PEAKreq = freq / [entry + (1 + read_ws) + (1 + write_ws) + exit]

where:

Table 25-7. Peak request formula operands

Operand	Description
PEAKreq	Peak request rate
freq	System frequency
entry	Channel startup (4 cycles)
read_ws	Wait states seen during the system bus read data phase
write_ws	Wait states seen during the system bus write data phase
exit	Channel shutdown (3 cycles)

25.4.4.3 eDMA performance example

Consider a system with the following characteristics:

- Internal SRAM can be accessed with one wait-state when viewed from the system bus data phase

- All internal peripheral bus reads require two wait-states, and internal peripheral bus writes three wait-states viewed from the system bus data phase
- System operates at 150 MHz

For an SRAM to internal peripheral bus transfer,

PEAKreq = 150 MHz / [4 + (1 + 1) + (1 + 3) + 3] cycles = 11.5 Mreq/sec

For an internal peripheral bus to SRAM transfer,

PEAKreq = 150 MHz / [4 + (1 + 2) + (1 + 1) + 3] cycles = 12.5 Mreq/sec

Assuming an even distribution of the two transfer types, the average peak request rate would be:

PEAKreq = (11.5 Mreq/sec + 12.5 Mreq/sec) / 2 = 12.0 Mreq/sec

The minimum number of cycles to perform a single read/write, zero wait states on the system bus, from a cold start where no channel is executing and eDMA is idle are:

- 11 cycles for a software, that is, a TCD*n*_CSR[START] bit, request
- 12 cycles for a hardware, that is, an eDMA peripheral request signal, request

Two cycles account for the arbitration pipeline and one extra cycle on the hardware request resulting from the internal registering of the eDMA peripheral request signals. For the peak request rate calculations above, the arbitration and request registering is absorbed in or overlaps the previous executing channel.

Note

When channel linking or scatter/gather is enabled, a two cycle delay is imposed on the next channel selection and startup. This allows the link channel or the scatter/gather channel to be eligible and considered in the arbitration pool for next channel selection.

25.5 Initialization/application information

The following sections discuss initialization of the eDMA and programming considerations.

25.5.1 eDMA initialization

To initialize the eDMA:

1. Write to the CR if a configuration other than the default is desired.
2. Write the channel priority levels to the DCHPRI*n* registers if a configuration other than the default is desired.
3. Enable error interrupts in the EEI register if so desired.
4. Write the 32-byte TCD for each channel that may request service.
5. Enable any hardware service requests via the ERQ register.
6. Request channel service via either:
   - Software: setting the TCD*n*_CSR[START]
   - Hardware: slave device asserting its eDMA peripheral request signal

After any channel requests service, a channel is selected for execution based on the arbitration and priority levels written into the programmer's model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in the following table, for the selected channel into its internal address path module.

As the TCD is read, the first transfer is initiated on the internal bus, unless a configuration error is detected. Transfers from the source, as defined by TCD*n*_SADDR, to the destination, as defined by TCD*n*_DADDR, continue until the number of bytes specified by TCD*n*_NBYTES are transferred.

When the transfer is complete, the eDMA engine's local TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_CITER are written back to the main TCD memory and any minor loop channel linking is performed, if enabled. If the major loop is exhausted, further post processing executes, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

Table 25-8. TCD Control and Status fields

TCD*n*_CSR field name	Description
START	Control bit to start channel explicitly when using a software initiated DMA service (Automatically cleared by hardware)
ACTIVE	Status bit indicating the channel is currently in execution
DONE	Status bit indicating major loop completion (cleared by software when using a software initiated DMA service)
D_REQ	Control bit to disable DMA request at end of major loop completion when using a hardware initiated DMA service
BWC	Control bits for throttling bandwidth control of a channel
E_SG	Control bit to enable scatter-gather feature

Table continues on the next page...

Table 25-8. TCD Control and Status fields (continued)

TCD*n*_CSR field name	Description
INT_HALF	Control bit to enable interrupt when major loop is half complete
INT_MAJ	Control bit to enable interrupt when major loop completes

The following figure shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, and one level of minor loop DMA preemption is allowed. The number of minor loops in a major loop is specified by the beginning iteration count (BITER).

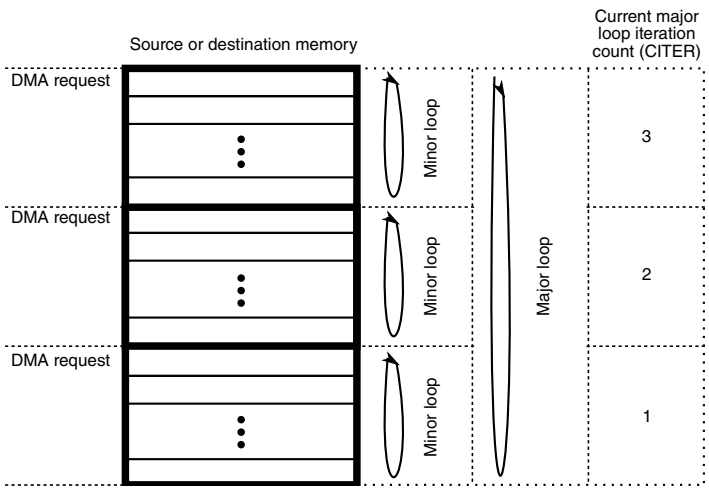


Figure 25-5. Example of multiple loop iterations

The following figure lists the memory array terms and how the TCD settings interrelate.

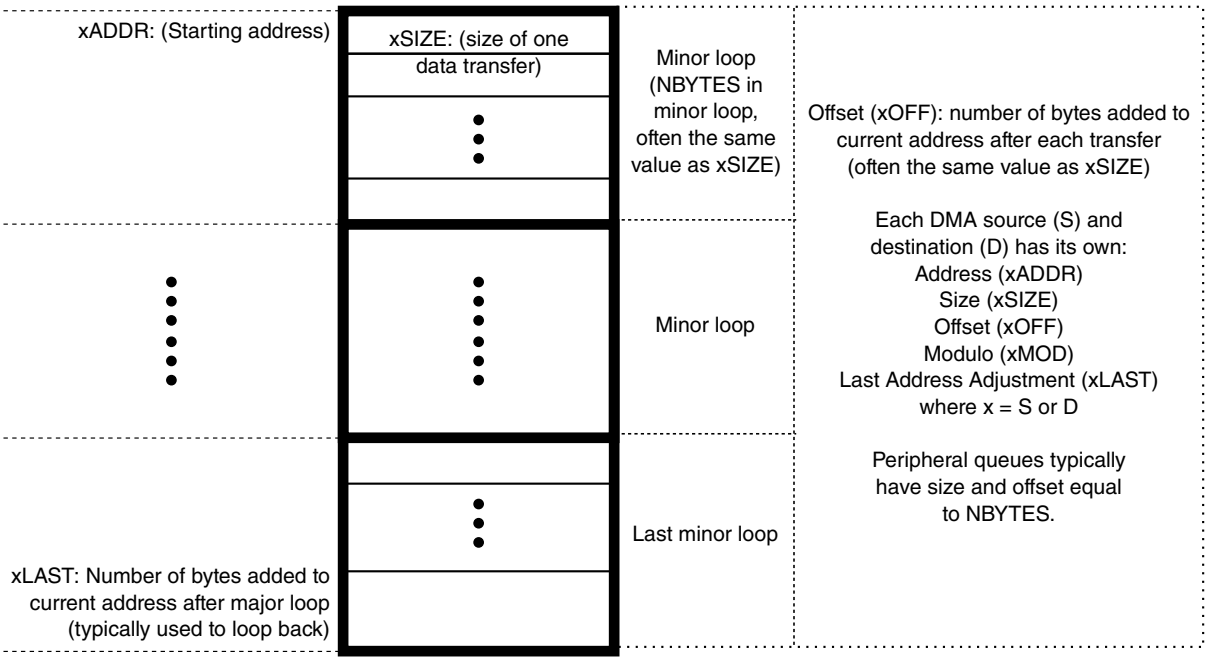


Figure 25-6. Memory array terms

25.5.2 Programming errors

The eDMA performs various tests on the transfer control descriptor to verify consistency in the descriptor data. Most programming errors are reported on a per channel basis with the exception of channel priority error (ES[CPE]).

For all error types other than channel priority error, the channel number causing the error is recorded in the Error Status register (DMAx_ES). If the error source is not removed before the next activation of the problem channel, the error is detected and recorded again.

If priority levels are not unique, when any channel requests service, a channel priority error is reported. The highest channel priority with an active request is selected, but the lowest numbered channel with that priority is selected by arbitration and executed by the eDMA engine. The hardware service request handshake signals, error interrupts, and error reporting is associated with the selected channel.

25.5.3 Arbitration mode considerations

This section discusses arbitration considerations for the eDMA.

25.5.3.1 Fixed channel arbitration

In this mode, the channel service request from the highest priority channel is selected to execute.

25.5.3.2 Round-robin channel arbitration

Channels are serviced starting with the highest channel number and rotating through to the lowest channel number without regard to the channel priority levels.

25.5.4 Performing DMA transfers

This section presents examples on how to perform DMA transfers with the eDMA.

25.5.4.1 Single request

To perform a simple transfer of n bytes of data with one activation, set the major loop to one (TCD*n*_CITER = TCD*n*_BITER = 1). The data transfer begins after the channel service request is acknowledged and the channel is selected to execute. After the transfer is complete, the TCD*n*_CSR[DONE] bit is set and an interrupt generates if properly enabled.

For example, the following TCD entry is configured to transfer 16 bytes of data. The eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has a byte wide memory port located at 0x1000. The destination memory has a 32-bit port located at 0x2000. The address offsets are programmed in increments to match the transfer size: one byte for the source and four bytes for the destination. The final source and destination addresses are adjusted to return to their beginning values.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA= -16
TCDn_CSR[INT_MAJ] = 1
TCDn_CSR[START] = 1 (Should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

This generates the following event sequence:

1. User write to the TCD*n*_CSR[START] bit requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
   a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
   b. Write 32-bits to location 0x2000 → first iteration of the minor loop.
   c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
   d. Write 32-bits to location 0x2004 → second iteration of the minor loop.
   e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
   f. Write 32-bits to location 0x2008 → third iteration of the minor loop.
   g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
   h. Write 32-bits to location 0x200C → last iteration of the minor loop → major loop complete.
6. The eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 1 (TCD*n*_BITER).
7. The eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0, TCD*n*_CSR[DONE] = 1, INT[*n*] = 1.
8. The channel retires and the eDMA goes idle or services the next channel.

25.5.4.2 Multiple requests

The following example transfers 32 bytes via two hardware requests, but is otherwise the same as the previous example. The only fields that change are the major loop iteration count and the final address offsets. The eDMA is programmed for two iterations of the major loop transferring 16 bytes per iteration. After the channel's hardware requests are enabled in the ERQ register, the slave device initiates channel service requests.

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

This would generate the following sequence of events:

1. First hardware, that is, eDMA peripheral, request for channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD*n* data from local memory to internal register file.
5. The source to destination transfers are executed as follows:
   a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
   b. Write 32-bits to location 0x2000 → first iteration of the minor loop.
   c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
   d. Write 32-bits to location 0x2004 → second iteration of the minor loop.
   e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
   f. Write 32-bits to location 0x2008 → third iteration of the minor loop.
   g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
   h. Write 32-bits to location 0x200C → last iteration of the minor loop.
6. eDMA engine writes: TCD*n*_SADDR = 0x1010, TCD*n*_DADDR = 0x2010, TCD*n*_CITER = 1.
7. eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0.
8. The channel retires → one iteration of the major loop. The eDMA goes idle or services the next channel.
9. Second hardware, that is, eDMA peripheral, requests channel service.
10. The channel is selected by arbitration for servicing.
11. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.

12. eDMA engine reads: channel TCD data from local memory to internal register file.
13. The source to destination transfers are executed as follows:
    a. Read byte from location 0x1010, read byte from location 0x1011, read byte from 0x1012, read byte from 0x1013.
    b. Write 32-bits to location 0x2010 → first iteration of the minor loop.
    c. Read byte from location 0x1014, read byte from location 0x1015, read byte from 0x1016, read byte from 0x1017.
    d. Write 32-bits to location 0x2014 → second iteration of the minor loop.
    e. Read byte from location 0x1018, read byte from location 0x1019, read byte from 0x101A, read byte from 0x101B.
    f. Write 32-bits to location 0x2018 → third iteration of the minor loop.
    g. Read byte from location 0x101C, read byte from location 0x101D, read byte from 0x101E, read byte from 0x101F.
    h. Write 32-bits to location 0x201C → last iteration of the minor loop → major loop complete.
14. eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 2 (TCD*n*_BITER).
15. eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0, TCD*n*_CSR[DONE] = 1, INT[n] = 1.
16. The channel retires → major loop complete. The eDMA goes idle or services the next channel.

25.5.4.3 Using the modulo feature

The modulo feature of the eDMA provides the ability to implement a circular data queue in which the size of the queue is a power of 2. MOD is a 5-bit field for the source and destination in the TCD, and it specifies which lower address bits increment from their original value after the address+offset calculation. All upper address bits remain the same as in the original value. A setting of 0 for this field disables the modulo feature.

The following table shows how the transfer addresses are specified based on the setting of the MOD field. Here a circular buffer is created where the address wraps to the original value while the 28 upper address bits (0x1234567*x*) retain their original value. In this example the source address is set to 0x12345670, the offset is set to 4 bytes and the MOD field is set to 4, allowing for a 2^4 byte (16-byte) size queue.

Table 25-9. Modulo example

Transfer Number	Address
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C
5	0x12345670
6	0x12345674

25.5.5 Monitoring transfer descriptor status

This section discusses how to monitor eDMA status.

25.5.5.1 Testing for minor loop completion

There are two methods to test for minor loop completion when using software initiated service requests. The first is to read the TCD*n*_CITER field and test for a change. Another method may be extracted from the sequence shown below. The second method is to test the TCD*n*_CSR[START] bit and the TCD*n*_CSR[ACTIVE] bit. The minor-loop-complete condition is indicated by both bits reading zero after the TCD*n*_CSR[START] was set. Polling the TCD*n*_CSR[ACTIVE] bit may be inconclusive, because the active status may be missed if the channel execution is short in duration.

The TCD status bits execute the following sequence for a software activated channel:

Stage	TCD*n*_CSR bits			State
	START	ACTIVE	DONE	
1	1	0	0	Channel service request via software
2	0	1	0	Channel is executing
3a	0	0	0	Channel has completed the minor loop and is idle
3b	0	0	1	Channel has completed the major loop and is idle

The best method to test for minor-loop completion when using hardware, that is, peripheral, initiated service requests is to read the TCD*n*_CITER field and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status bits execute the following sequence for a hardware-activated channel:

Stage	TCD*n*_CSR bits			State
	START	ACTIVE	DONE	
1	0	0	0	Channel service request via hardware (peripheral request asserted)
2	0	1	0	Channel is executing
3a	0	0	0	Channel has completed the minor loop and is idle
3b	0	0	1	Channel has completed the major loop and is idle

For both activation types, the major-loop-complete status is explicitly indicated via the TCD*n*_CSR[DONE] bit.

The TCD*n*_CSR[START] bit is cleared automatically when the channel begins execution regardless of how the channel activates.

25.5.5.2 Reading the transfer descriptors of active channels

The eDMA reads back the true TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_NBYTES values if read while a channel executes. The true values of the SADDR, DADDR, and NBYTES are the values the eDMA engine currently uses in its internal register file and not the values in the TCD local memory for that channel. The addresses, SADDR and DADDR, and NBYTES, which decrement to zero as the transfer progresses, can give an indication of the progress of the transfer. All other values are read back from the TCD local memory.

25.5.5.3 Checking channel preemption status

Preemption is available only when fixed arbitration is selected as the channel arbitration mode. A preemptive situation is one in which a preempt-enabled channel runs and a higher priority request becomes active. When the eDMA engine is not operating in fixed channel arbitration mode, the determination of the actively running relative priority outstanding requests become undefined. Channel priorities are treated as equal, that is, constantly rotating, when Round-Robin Arbitration mode is selected.

The TCD*n*_CSR[ACTIVE] bit for the preempted channel remains asserted throughout the preemption. The preempted channel is temporarily suspended while the preempting channel executes one major loop iteration. If two TCD*n*_CSR[ACTIVE] bits are set simultaneously in the global TCD map, a higher priority channel is actively preempting a lower priority channel.

25.5.6 Channel Linking

Channel linking (or chaining) is a mechanism where one channel sets the TCD*n*_CSR[START] bit of another channel (or itself), therefore initiating a service request for that channel. When properly enabled, the EDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs at the completion of the minor loop (or one iteration of the major loop). The TCD*n*_CITER[E_LINK] field determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. When the major loop is exhausted, only the major loop channel link fields are used to determine if a channel link should be made. For example, the initial fields of:

```
TCDn_CITER[E_LINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJOR_E_LINK] = 1
TCDn_CSR[MAJOR_LINKCH] = 0x7
```

executes as:

1. Minor loop done → set TCD12_CSR[START] bit
2. Minor loop done → set TCD12_CSR[START] bit
3. Minor loop done → set TCD12_CSR[START] bit
4. Minor loop done, major loop done→ set TCD7_CSR[START] bit

When minor loop linking is enabled (TCD*n*_CITER[E_LINK] = 1), the TCD*n*_CITER[CITER] field uses a nine bit vector to form the current iteration count. When minor loop linking is disabled (TCD*n*_CITER[E_LINK] = 0), the TCD*n*_CITER[CITER] field uses a 15-bit vector to form the current iteration count. The bits associated with the TCD*n*_CITER[LINKCH] field are concatenated onto the CITER value to increase the range of the CITER.

Note

The TCD*n*_CITER[E_LINK] bit and the TCD*n*_BITER[E_LINK] bit must equal or a configuration error is reported. The CITER and BITER vector widths must be equal to calculate the major loop, half-way done interrupt point.

The following table summarizes how a DMA channel can link to another DMA channel, i.e, use another channel's TCD, at the end of a loop.

Table 25-10. Channel Linking Parameters

Desired Link Behavior	TCD Control Field Name	Description
Link at end of Minor Loop	CITER[E_LINK]	Enable channel-to-channel linking on minor loop completion (current iteration)
	CITER[LINKCH]	Link channel number when linking at end of minor loop (current iteration)
Link at end of Major Loop	CSR[MAJOR_E_LINK]	Enable channel-to-channel linking on major loop completion
	CSR[MAJOR_LINKCH]	Link channel number when linking at end of major loop

25.5.7 Dynamic programming

This section provides recommended methods to change the programming model during channel execution.

25.5.7.1 Dynamically changing the channel priority

The following two options are recommended for dynamically changing channel priority levels:

1. Switch to Round-Robin Channel Arbitration mode, change the channel priorities, then switch back to Fixed Arbitration mode,
2. Disable all the channels, change the channel priorities, then enable the appropriate channels.

25.5.7.2 Dynamic channel linking

Dynamic channel linking is the process of setting the TCDn_CSR[MAJORELINK] bit during channel execution (see the diagram in TCD structure). This bit is read from the TCD local memory at the end of channel execution, thus allowing the user to enable the feature during channel execution.

Because the user is allowed to change the configuration during execution, a coherency model is needed. Consider the scenario where the user attempts to execute a dynamic channel link by enabling the TCDn_CSR[MAJORELINK] bit at the same time the eDMA engine is retiring the channel. The TCDn_CSR[MAJORELINK] would be set in the programmer's model, but it would be unclear whether the actual link was made before the channel retired.

The following coherency model is recommended when executing a dynamic channel link request.

1. Write 1 to the TCDn_CSR[MAJORELINK] bit.
2. Read back the TCDn_CSR[MAJORELINK] bit.
3. Test the TCDn_CSR[MAJORELINK] request status:
   - If TCDn_CSR[MAJORELINK] = 1, the dynamic link attempt was successful.
   - If TCDn_CSR[MAJORELINK] = 0, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces the TCDn_CSR[MAJORELINK] bit to zero on any writes to a channel's TCD.word7 after that channel's TCD.done bit is set, indicating the major loop is complete.

NOTE

The user must clear the The TCDn_CSR[DONE] bit before writing the TCDn_CSR[MAJORELINK] bit. The TCDn_CSR[DONE] bit is cleared automatically by the eDMA engine after a channel begins execution.

25.5.7.3 Dynamic scatter/gather

Scatter/gather is the process of automatically loading a new TCD into a channel. It allows a DMA channel to use multiple TCDs; this enables a DMA channel to scatter the DMA data to multiple destinations or gather it from multiple sources. When scatter/gather is enabled and the channel has finished its major loop, a new TCD is fetched from system memory and loaded into that channel's descriptor location in eDMA programmer's model, thus replacing the current descriptor.

Because the user is allowed to change the configuration during execution, a coherency model is needed. Consider the scenario where the user attempts to execute a dynamic scatter/gather operation by enabling the TCDn_CSR[ESG] bit at the same time the eDMA engine is retiring the channel. The ESG bit would be set in the programmer's model, but it would be unclear whether the actual scatter/gather request was honored before the channel retired.

Two methods for this coherency model are shown in the following subsections. Method 1 has the advantage of reading the MAJORLINKCH field and the ESG bit with a single read. For both dynamic channel linking and scatter/gather requests, the TCD local memory controller forces the TCD MAJOR.E_LINK and E_SG bits to zero on any writes to a channel's TCD word 7 if that channel's TCD.DONE bit is set indicating the major loop is complete.

NOTE

The user must clear the TCDn_CSR[DONE] bit before writing the MAJORELINK or ESG bits. The TCDn_CSR[DONE] bit is cleared automatically by the eDMA engine after a channel begins execution.

25.5.7.3.1 Method 1 (channel not using major loop channel linking)

For a channel not using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request.

When the TCDn_CSR[MAJORELINK] bit is zero, the TCDn_CSR[MAJORLINKCH] field is not used by the eDMA. In this case, the MAJORLINKCH field may be used for other purposes. This method uses the MAJORLINKCH field as a TCD identification (ID).

1. When the descriptors are built, write a unique TCD ID in the TCDn_CSR[MAJORLINKCH] field for each TCD associated with a channel using dynamic scatter/gather.
2. Write 1b to the TCDn_CSR[DREQ] bit.

   Should a dynamic scatter/gather attempt fail, setting the DREQ bit will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (DADDR) that was calculated using a scatter/gather address (written in the next step) instead of a dlast final offset value.
3. Write the TCDn_DLASTSGA register with the scatter/gather address.
4. Write 1b to the TCDn_CSR[ESG] bit.

5. Read back the 16 bit TCD control/status field.
6. Test the ESG request status and MAJORLINKCH value in the TCDn_CSR register:

   If ESG = 1b, the dynamic link attempt was successful.

   If ESG = 0b and the MAJORLINKCH (ID) did not change, the attempted dynamic link did not succeed (the channel was already retiring).

   If ESG = 0b and the MAJORLINKCH (ID) changed, the dynamic link attempt was successful (the new TCD's E_SG value cleared the ESG bit).

25.5.7.3.2 Method 2 (channel using major loop channel linking)

For a channel using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request. This method uses the TCD.DLAST_SGA field as a TCD identification (ID).

1. Write 1b to the TCDn_CSR[DREQ] bit.

   Should a dynamic scatter/gather attempt fail, setting the DREQ bit will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (DADDR) that was calculated using a scatter/gather address (written in the next step) instead of a dlast final offset value.
2. Write the TCDn_DLASTSGA register with the scatter/gather address.
3. Write 1b to the TCDn_CSR[ESG] bit.
4. Read back the ESG bit.
5. Test the ESG request status:

   If ESG = 1b, the dynamic link attempt was successful.

   If ESG = 0b, read the 32 bit TCDn_DLASTSGA field.

   If ESG = 0b and the TCDn_DLASTSGA did not change, the attempted dynamic link did not succeed (the channel was already retiring).

   If ESG = 0b and the TCDn_DLASTSGA changed, the dynamic link attempt was successful (the new TCD's E_SG value cleared the ESG bit).

25.5.8 Suspend/resume a DMA channel with active hardware service requests

The DMA allows the user to move data from memory or peripheral registers to another location in memory or peripheral registers without CPU interaction. Once the DMA and peripherals have been configured and are active, it is rare to suspend a peripheral's service request dynamically. In this scenario, there are certain restrictions to disabling a DMA hardware service request. For coherency, a specific procedure must be followed. This section provides guidance on how to coherently suspend and resume a Direct Memory Access (DMA) channel when the DMA is triggered by a slave module such as the Serial Peripheral Interface (DSPI), ADC, or other module.

25.5.8.1 Suspend an active DMA channel

To suspend an active DMA channel:

1. Stop the DMA service request at the peripheral first. Confirm it has been disabled by reading back the appropriate register in the peripheral.
2. Check the DMA's Hardware Request Status Register (DMA_HRSn) to ensure there is no service request to the DMA channel being suspended. Then disable the hardware service request by clearing the ERQ bit on appropriate DMA channel.

25.5.8.2 Resume a DMA channel

To resume a DMA channel:

1. Enable the DMA service request on the appropriate channel by setting the its ERQ bit.
2. Enable the DMA service request at the peripheral.

For example, assume the DSPI is set as a master for transmitting data via a DMA service request when the DSPI_TXFIFO has an empty slot. The DMA will transfer the next command and data to the TXFIFO upon the request. If the user needs to suspend the DMA/DSPI transfer loop, perform the following steps:

1. Disable the DMA service request at the source by writing 0 to DSPI_RSER[TFFF_RE] . Confirm that DSPI_RSER[TFFF_RE] is 0.
2. Ensure there is no DMA service request from the DSPI by verifying that DMA_HRS[HRS*n*] is 0 for the appropriate channel. If no service request is present, disable the DMA channel by clearing the channel's ERQ bit. If a service request is present, wait until the request has been processed and the HRS bit reads zero.

Chapter 26
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.

26.1 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.

26.1.1 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.

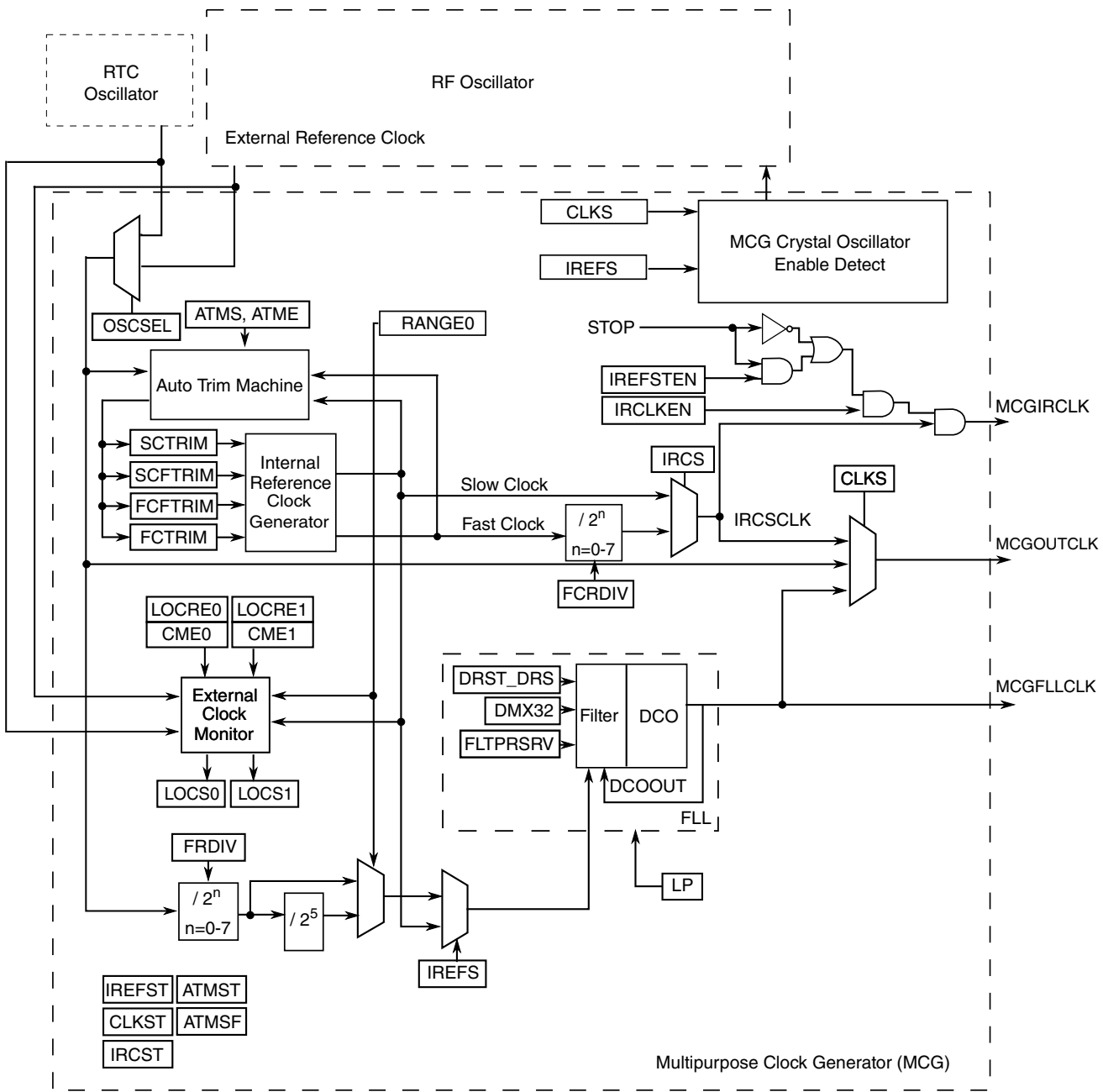


Figure 26-1. Multipurpose Clock Generator (MCG) block diagram

26.1.2 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.

26.2 External Signal Description

There are no MCG signals that connect off chip.

26.3 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	26.3.1/573
4006_4001	MCG Control 2 Register (MCG_C2)	8	R/W	See section	26.3.2/574
4006_4002	MCG Control 3 Register (MCG_C3)	8	R/W	Undefined	26.3.3/575
4006_4003	MCG Control 4 Register (MCG_C4)	8	R/W	See section	26.3.4/576
4006_4004	MCG Control 5 Register (MCG_C5)	8	R/W	00h	26.3.5/577
4006_4005	MCG Control 6 Register (MCG_C6)	8	R/W	00h	26.3.5/577
4006_4006	MCG Status Register (MCG_S)	8	R	10h	26.3.6/578
4006_4008	MCG Status and Control Register (MCG_SC)	8	R/W	02h	26.3.7/579
4006_400A	MCG Auto Trim Compare Value High Register (MCG_ATCVH)	8	R/W	00h	26.3.8/580
4006_400B	MCG Auto Trim Compare Value Low Register (MCG_ATCVL)	8	R/W	00h	26.3.9/581
4006_400C	MCG Control 7 Register (MCG_C7)	8	R/W	00h	26.3.10/ 581
4006_400D	MCG Control 8 Register (MCG_C8)	8	R/W	See section	26.3.11/ 582
4006_4011	MCG Control 12 Register (MCG_C12)	8	R/W	00h	26.3.12/ 583
4006_4012	MCG Status 2 Register (MCG_S2)	8	R/W	00h	26.3.12/ 583
4006_4013	MCG Test 3 Register (MCG_T3)	8	R/W	00h	26.3.12/ 583

26.3.1 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.

Table continues on the next page...

MCG_C1 field descriptions (continued)

Field	Description
	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.

26.3.2 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	x*	0	0	0	0	0	0

* Notes:
- x = Undefined at reset.

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.

Table continues on the next page...

MCG_C2 field descriptions (continued)

Field	Description
	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)
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.

26.3.3 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.

26.3.4 MCG Control 4 Register (MCG_C4)

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.

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.

NOTE: 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 rowspan="2">00</td><td>0</td><td>31.25–39.0625 kHz</td><td>640</td><td>20–25 MHz</td></tr>
<tr><td>1</td><td>32.768 kHz</td><td>732</td><td>24 MHz</td></tr>
<tr><td rowspan="2">01</td><td>0</td><td>31.25–39.0625 kHz</td><td>1280</td><td>40–50 MHz</td></tr>
<tr><td>1</td><td>32.768 kHz</td><td>1464</td><td>48 MHz</td></tr>
<tr><td rowspan="2">10</td><td>0</td><td>31.25–39.0625 kHz</td><td>1920</td><td>60–75 MHz</td></tr>
<tr><td>1</td><td>32.768 kHz</td><td>2197</td><td>72 MHz</td></tr>
<tr><td rowspan="2">11</td><td>0</td><td>31.25–39.0625 kHz</td><td>2560</td><td>80–100 MHz</td></tr>
<tr><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.

00 Encoding 0 — Low range (reset default).
01 Encoding 1 — Mid range.
10 Encoding 2 — Mid-high range.
11 Encoding 3 — High range.</td></tr>
<tr><td>4–1
FCTRIM</td><td>Fast Internal Reference Clock Trim Setting</td></tr>
</table>

Table continues on the next page...

MCG_C4 field descriptions (continued)

Field	Description
	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.

26.3.5 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.

26.3.5 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

Table continues on the next page...

MCG_C6 field descriptions (continued)

Field	Description
	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.

26.3.6 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. 00 Encoding 0 — Output of the FLL is selected (reset default). 01 Encoding 1 — Internal reference clock is selected.

Table continues on the next page...

MCG_S field descriptions (continued)

Field	Description
	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).

26.3.7 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.
5 ATMF	Automatic Trim Machine Fail Flag

Table continues on the next page...

MCG_SC field descriptions (continued)

Field	Description
	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.

26.3.8 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.

26.3.9 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.

26.3.10 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.

26.3.11 MCG Control 8 Register (MCG_C8)

Address: 4006_4000h base + Dh offset = 4006_400Dh

Bit	7	6	5	4	3	2	1	0
Read	LOCRE1	0	CME1	0				LOCS1
Write								w1c
Reset	1	0	0	0	0	0	0	0

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

26.3.12 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.

26.3.12 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.

26.3.12 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.

26.4 Functional description

26.4.1 MCG mode state diagram

The seven states of the MCG are shown in the following figure and are described in Table 26-1. The arrows indicate the permitted MCG mode transitions.

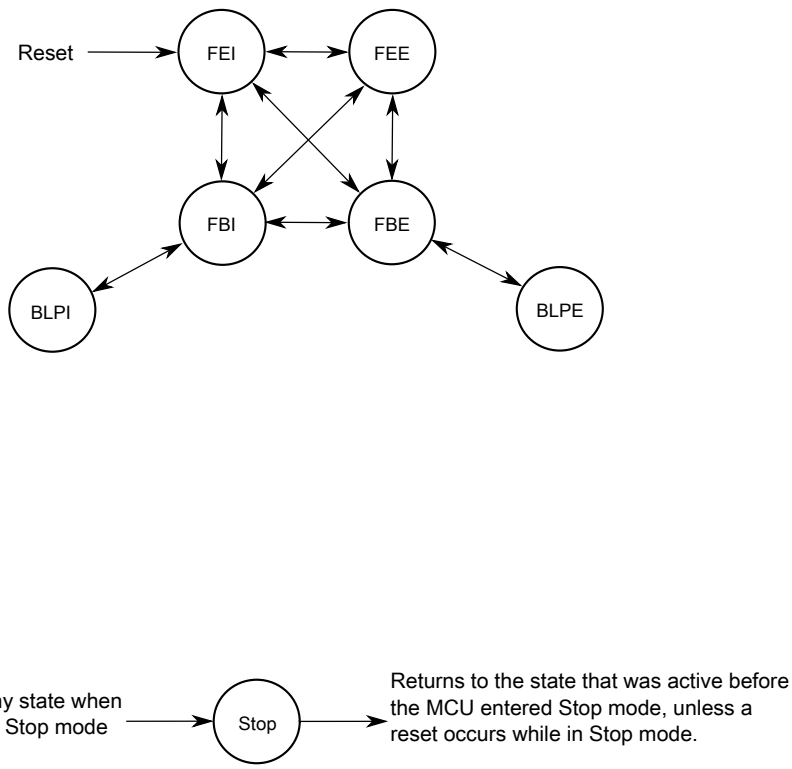


Figure 26-2. MCG mode state diagram

26.4.1.1 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 26-2.

Table 26-1. 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) [1]	Bypassed Low Power Internal (BLPI) mode is entered when all the following conditions occur:

Table continues on the next page...

Table 26-1. 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) [1]	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 Power management 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)

1. **Caution:** If entering VLPR mode, MCG has to be configured and enter BLPE mode or BLPI mode with the Fast IRC clock selected (C2[IRCS]=1). After it enters VLPR mode, writes to any of the MCG control registers that can cause an MCG clock mode switch to a non low power clock mode must be avoided.

NOTE

For the chip-specific modes of operation, see the power management chapter of this MCU.

26.4.1.2 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 $t_{fll\_acquire}$.

26.4.2 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].

26.4.3 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.

26.4.3.1 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 FCRDIV 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.

26.4.4 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.

26.4.5 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.

$$\text{Expected Count Value} = (Fe / Fr) * 21 * (128)$$

26.5 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.

26.5.1 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.

26.5.1.1 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 26-2). 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.

26.5.2 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, potentially resulting in unpredictable behavior.

26.5.3 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 26-2. MCGOUTCLK Frequency Calculation Options

Clock Mode	$f_{MCGOUTCLK}$[1]	Note
FEI (FLL engaged internal)	$f_{int} \times F$	Typical $f_{MCGOUTCLK}$ = 21 MHz immediately after reset.
FEE (FLL engaged external)	$(f_{ext} / FLL\_R) \times F$	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 27
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.

27.1 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.

27.1.1 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 .

27.1.2 Block Diagram

The following is the block diagram of the RTC oscillator.

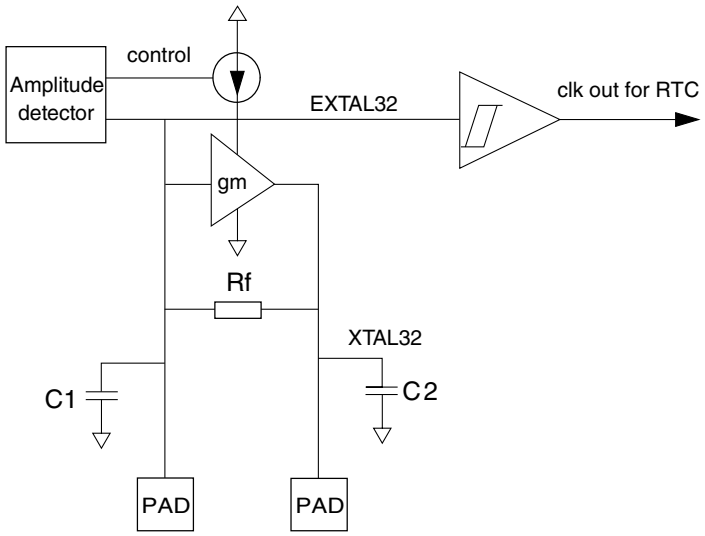


Figure 27-1. RTC Oscillator Block Diagram

27.2 RTC Signal Descriptions

The following table shows the user-accessible signals available for the RTC oscillator. See the Pinouts or Signal Multiplexing and Pin Assignment section to find out which signals are actually connected to the external pins.

Table 27-1. RTC Signal Descriptions

Signal	Description	I/O
EXTAL32	Oscillator Input	I
XTAL32	Oscillator Output	O

27.2.1 EXTAL32 — Oscillator Input

This signal is the analog input of the RTC oscillator.

27.2.2 XTAL32 — Oscillator Output

This signal is the analog output of the RTC oscillator module.

27.3 External Crystal Connections

The connections with a crystal is shown in the following figure. External load capacitors and feedback resistor are not required.

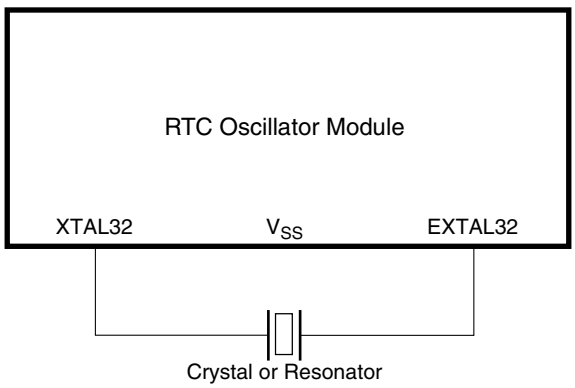


Figure 27-2. Crystal Connections

27.4 Memory Map/Register Descriptions

RTC oscillator control bits are part of the RTC registers. Refer to RTC Control Register (RTC_CR) in the RTC chapter, or RTC_GP_DATA_REG (if this General Purpose Data Register exists in the RTC registers) description in the chip-specific information section, for more details.

27.5 Functional Description

As shown in Figure 27-1, 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.

27.6 Reset Overview

There is no reset state associated with the RTC oscillator.

27.7 Interrupts

The RTC oscillator does not generate any interrupts.

Chapter 28
Flash Memory Controller (FMC)

28.1 Introduction

The Flash Memory Controller (FMC) is a memory interface and acceleration unit, which provides:

- an interface between the device and the dual-bank nonvolatile memory. Bank 0 consists of program flash memory, and bank 1 consists of FlexNVM.
- buffers that can accelerate flash memory and FlexNVM data transfers.

28.1.1 Overview

The Flash Memory Controller manages the interface between the device and the dual-bank flash memory. The FMC receives status information describing the configuration of the memory and uses this information to ensure a proper interface. The next table shows the supported read/write operations.

Flash memory type	Read	Write
Program flash memory	8-bit, 16-bit, and 32-bit reads	—[1]
FlexNVM used as Data flash memory	8-bit, 16-bit, and 32-bit reads	—[1]
FlexNVM and FlexRAM used as EEPROM	8-bit, 16-bit, and 32-bit reads	8-bit, 16-bit, and 32-bit writes

1. A write operation to program flash memory or to FlexNVM used as data flash memory results in a bus error.

- The FMC's input bus can actually operate faster than the flash memory.
- The FMC-to-flash interface has flow control, and can add wait states as needed, for input bus reads that need flash accesses.
- The FMC also contains various configurable buffers that hold recent flash accesses. If an input bus read hits a valid buffer, that access will finish with no wait states.

28.1.2 Features

- Interface between the device and the dual-bank flash memory and FlexMemory:
  - The FMC's input bus supports 8-bit, 16-bit, and 32-bit read operations to flash memory and FlexNVM used as data flash memory.
  - The FMC's flash memory interface fetches a {32, 64, 128, 256}-bit flash page.
  - For input read requests, the FMC fetches a flash page with the desired read data from flash memory.
- Acceleration of data transfer from flash memory and FlexMemory to the device:
  - A flash-page-sized buffer that holds the current flash page fetched due to a FMC read request. Subsequent FMC read requests *that hit in the current buffer* return data with no wait states.
  - A flash-page-sized prefetch speculation buffer with controls for prefetching on instructions and/or data reads. When prefetching is enabled, idle FMC-to-flash interface cycles are used to fetch the next sequential flash page and hold it in the prefetch buffer. Subsequent FMC read requests *that hit in the speculation buffer* will return data with no wait states.
  - Input controls:
    - to disable data type speculation
    - to disable all speculation
    - to invalidate the current and speculation buffers
  - The flash cache has input controls:
    - to disable instruction caching
    - to disable operand caching
    - to disable all caching
    - to clear the cache
  - The size of the flash cache in bytes is calculated as follows:

    ```
    flash cache size = [number of ways] x [number of sets] x [flash page size (in bytes)]
    ```

    For example, a flash cache with 4 ways, 4 sets, and a 128-bit flash page (= 16 bytes) has a total flash cache size = 256 bytes

    (4 ways) x (4 sets) x (16 bytes per flash page) = 256 bytes, the size of the flash cache

28.2 Modes of operation

The FMC only operates when a bus master accesses the flash memory or FlexMemory.

For any device power mode where the flash memory or FlexMemory cannot be accessed, the FMC is disabled.

28.3 External signal description

The FMC has no external signals.

28.4 Functional description

The FMC is a flash interface and acceleration unit, with flexible buffers for user configuration.

- The FMC's input bus can operate faster than the flash memory.
- The FMC-to-flash interface has flow control to add wait states as needed (for input bus reads that need flash accesses).
- The FMC also contains various configurable buffers that hold recent flash accesses. If an input bus read hits a valid buffer, then that access will complete with no wait states.

28.4.1 Default configuration

After system reset, the FMC is configured to provide a significant level of buffering for transfers from the flash memory or FlexMemory:

- These masters have write access to a portion of bank 1 when FlexNVM is used with FlexRAM as EEPROM.
- For all banks:
  - The current and speculation buffers are cleared by reset.
  - Prefetch support for data and instructions is enabled.
  - The cache is cleared by reset.
  - The cache is configured for data or instruction replacement.

28.4.2 Configuration options

The default configuration provides a high degree of flash acceleration, however, advanced users may want to customize the FMC buffer configurations, to maximize throughput for their use cases. When reconfiguring the FMC for custom use cases, do not program the FMC's control registers while the flash memory or FlexMemory is being accessed. Instead, change the control registers with a routine executing from RAM in supervisor mode.

The FMC's cache and buffering controls and PFB1CR allow the tuning of resources to suit specific application requirements. The cache and buffer are each controlled individually. The controls enable buffering and prefetching per memory bank and access type (instruction fetch or data reference).

As an application example: if both instruction fetches and data references are accessing flash memory, then control is available to send instruction fetches, data references, or both to the cache or the single-entry buffer. Likewise, speculation can be enabled or disabled for either type of access.

In another application example, the cache can be configured for replacement from bank 0, while the single-entry buffer can be enabled for bank 1 only. This configuration is ideal for applications that use bank 0 for program space and bank 1 for data space.

28.4.3 Wait states

Because the core, crossbar switch, and bus masters can be clocked at a higher frequency than the flash clock, flash memory accesses that do not hit in the speculation buffer or cache usually require wait states.

All wait states and synchronization delays are handled automatically by the Flash Memory Controller. No direct user configuration is required (or even allowed) to set up the flash wait states.

28.4.4 Speculative reads

The FMC has a single buffer that reads ahead to the next page in the flash memory if there is an idle cycle. Speculative prefetching is programmable for instruction and/or data accesses. Because many code accesses are sequential, using the speculative prefetch buffer improves performance in most cases.

When speculative reads are enabled, the FMC immediately requests the next sequential page address after a read completes. By requesting the next page immediately, speculative reads can help to reduce or even eliminate wait states when accessing sequential code and/or data.

28.4.5 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

28.4.5.1 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 28-1. 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

Table continues on the next page...

Table 28-1. NVM Locations (continued)

NVM location	Description	
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 28-2. 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.

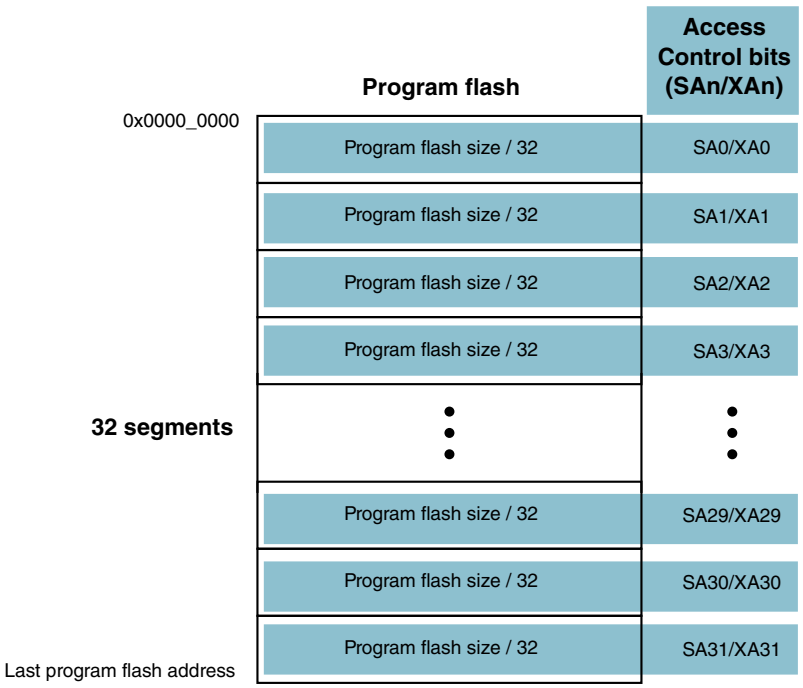


Figure 28-1. Program flash protection (32 segments)

28.4.5.1.1 Interface Signals

Table 28-3. 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.

28.4.5.1.2 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 command 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.

28.4.5.1.3 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.....

Table continues on the next page...

	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.

28.4.5.1.4 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, and may require 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 that one execute-only segment can read the execute-only code in another segment. Therefore, if we at the factory are sending pre-loaded code to another vendor, then that vendor will have access to our factory code. NDAs and legal agreements might help deal with this issue.
- **For single pre-loads** (for example, if we at the factory are pre-loading for a general purpose (GP) market or if a vendor with a blank part is pre-loading their 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.

28.4.5.1.5 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()
```

28.4.5.1.6 FAC application tips

In one use case, the NVSACC1 and NVXACC1 locations are programmed by NXP and they protect NXP 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.

28.5 Initialization and application information

The FMC does not require user initialization. Flash acceleration features are enabled by default.

The FMC has no visibility into flash memory erase and program cycles because the Flash Memory module manages them directly. As a result, if an application is executing flash memory commands, the FMC's current buffer, speculation buffer, and cache might need to be disabled and/or flushed to prevent the possibility of returning stale data.

Chapter 29
Flash Memory Module (FTFE)

29.1 Introduction

The FTFE module includes the following accessible memory regions:

- Program flash memory for vector space and code store
- For FlexNVM devices: FlexNVM for data store and additional code store
- For FlexNVM devices: FlexRAM for high-endurance data store or traditional RAM
- For program flash only devices: Programming acceleration RAM to speed flash programming

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 FTFE 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.

29.1.1 Features

The FTFE module includes the following features.

NOTE

See Memories and Memory Interfaces chapter for the exact amount of flash memory available on your device.

29.1.1.1 Program Flash Memory Features

- Sector size of 2 Kbytes
- 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
- Section programming for faster bulk programming times
- For devices containing only program flash memory: Read access to one program flash block is possible while programming or erasing data in another program flash block
- For devices containing FlexNVM memory: Read access to the program flash block is possible while programming or erasing data in the data flash block or FlexRAM

29.1.1.2 FlexNVM memory features

When FlexNVM is partitioned for data flash memory (on devices that contain FlexNVM memory):

- Sector size of 2 Kbytes
- Protection scheme prevents accidental program or erase of stored data
- Automated, built-in program and erase algorithms with verify
- Section programming for faster bulk programming times
- Read access to the data flash block possible while programming or erasing data in the program flash block

29.1.1.3 Programming Acceleration RAM features

- For devices with only program flash memory: RAM to support section programming

29.1.1.4 FlexRAM features

For devices with FlexNVM memory:

- Memory that can be used as traditional RAM or as high-endurance EEPROM storage
- Up to 8 Kbytes of FlexRAM configured for EEPROM or traditional RAM operations
- When configured for EEPROM:
  - Protection scheme prevents accidental program or erase of data written for EEPROM
  - Built-in hardware emulation scheme to automate EEPROM record maintenance functions
  - Programmable EEPROM data set size and FlexNVM partition code facilitating EEPROM memory endurance trade-offs
  - Supports FlexRAM aligned writes of 1, 2, or 4 bytes at a time
  - Read access to FlexRAM possible while programming or erasing data in the program or data flash memory
- When configured for traditional RAM:
  - Read and write access possible to the FlexRAM while programming or erasing data in the program or data flash memory

29.1.1.5 Other FTFE 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

29.1.2 Block diagram

The block diagram of the FTFE module is shown in the following figure.

For devices with FlexNVM feature:

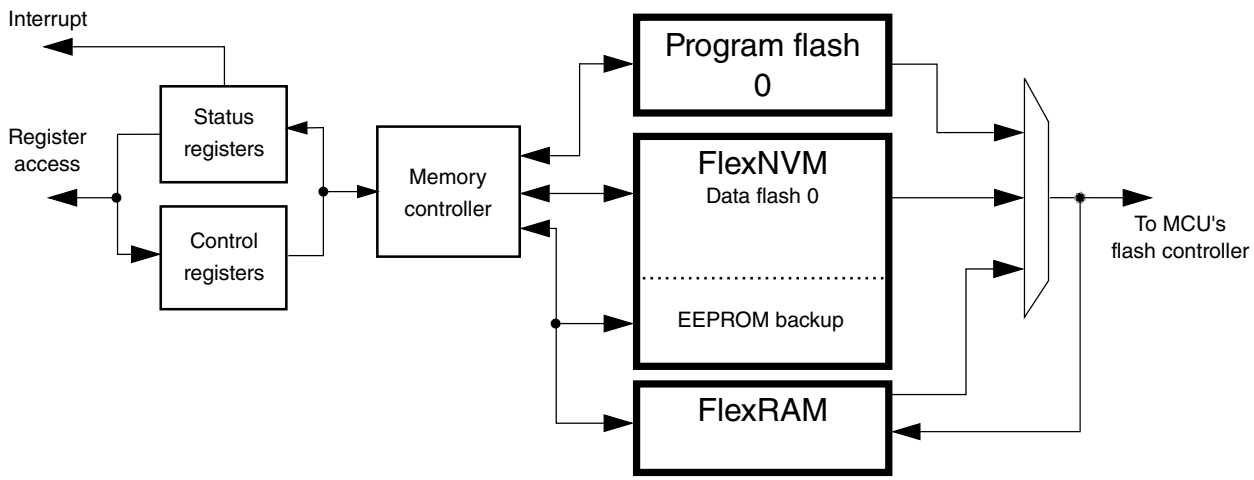


Figure 29-1. FTFE block diagram

For devices that contain only program flash:

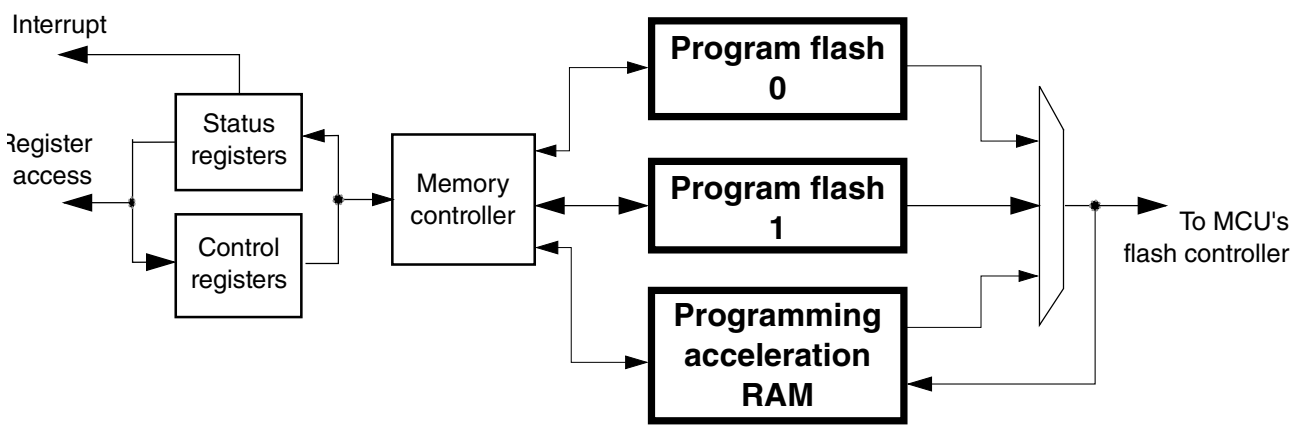


Figure 29-2. FTFE block diagram

29.1.3 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 FTFE module.

Data flash memory — Partitioned from the FlexNVM block, the data flash memory provides nonvolatile storage for user data, boot code, and additional code store.

Data flash sector — The data flash sector is the smallest portion of the data flash memory that can be erased.

EEPROM — Using a built-in filing system, the FTFE module emulates the characteristics of an EEPROM by effectively providing a high-endurance, byte-writeable (program and erase) NVM.

EEPROM backup data header — The EEPROM backup data header is comprised of a 64-bit field found in EEPROM backup data memory which contains information used by the EEPROM filing system to determine the status of a specific EEPROM backup flash sector.

EEPROM backup data record — The EEPROM backup data record is comprised of a 7-bit status field, a 13-bit address field, and a 32-bit data field found in EEPROM backup data memory which is used by the EEPROM filing system. If the status field indicates a record is valid, the data field is mirrored in the FlexRAM at a location determined by the address field.

EEPROM backup data memory — Partitioned from the FlexNVM block, EEPROM backup data memory provides nonvolatile storage for the EEPROM filing system representing data written to the FlexRAM requiring highest endurance.

EEPROM backup data sector — The EEPROM backup data sector contains one EEPROM header and up to 255 EEPROM backup data records, which are used by the EEPROM filing system.

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 FTFE module.

Flash block — A macro within the FTFE module which provides the nonvolatile memory storage.

FlexMemory — FTFE configuration that supports data flash, EEPROM, and FlexRAM.

FlexNVM Block — The FlexNVM block can be configured to be used as data flash memory, EEPROM backup flash memory, or a combination of both.

FlexRAM — The FlexRAM refers to a RAM, dedicated to the FTFE module, that can be configured to store EEPROM data or as traditional RAM. When configured for EEPROM, valid writes to the FlexRAM generates a new EEPROM backup data record stored in the EEPROM backup flash memory.

FTFE 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.

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 FTFE 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 FTFE module.

Phrase — 64 bits of data with an aligned phrase having byte-address[2:0] = 000.

Longword — 32 bits of data with an aligned longword having byte-address[1:0] = 00.

Word — 16 bits of data with an aligned word having byte-address[0] = 0.

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.

Section program buffer — Lower quarter of the programming acceleration FlexRAM allocated for storing large amounts of data for programming via the Program Section command.

Secure — An MCU state conveyed to the FTFE module as described in the Chip Configuration details for this device. In the secure state, reading and changing NVM contents is restricted.

29.2 External signal description

The FTFE module contains no signals that connect off-chip.

29.3 Memory map and registers

This section describes the memory map and registers for the FTFE module. Data read from unimplemented memory space in the FTFE module is undefined. Writes to unimplemented or reserved memory space (registers) in the FTFE module are ignored.

29.3.1 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 FTFE module.

NOTE

The flash configuration field offset addresses are relative byte addresses. Check your device specific memory map for the location of the program flash memory.

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 MCU 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	Program flash only devices: Reserved FlexNVM devices: Data flash protection byte. Refer to the description of the Data Flash Protection Register (FDPROT).
0x0_040E	1	Program flash only devices: Reserved

Table continues on the next page...

Flash Configuration Field Offset Address	Size (Bytes)	Field Description
		FlexNVM devices: EEPROM protection byte. Refer to the description of the EEPROM Protection Register (FEPROT).
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).

29.3.2 Program flash 0 IFR map

The program flash 0 IFR is a 1 Kbyte 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 0 IFR are summarized in the following table and further described in the subsequent paragraphs.

The program flash 0 IFR is located within the program flash 0 memory block.

Address Range	Size (Bytes)	Field Description
0x000 – 0x39F	928	Reserved
0x3A0 – 0x3A3	4	Program Once XACCH-1 Field (index = 0x08)
0x3A4 – 0x3A7	4	Program Once XACCL-1 Field (index = 0x08)
0x3A8 – 0x3AB	4	Program Once XACCH-2 Field (index = 0x09)
0x3AC – 0x3AF	4	Program Once XACCL-2 Field (index = 0x09)
0x3B0 – 0x3B3	4	Program Once SACCH-1 Field (index = 0x0A)
0x3B4 – 0x3B7	4	Program Once SACCL-1 Field (index = 0x0A)
0x3B8 – 0x3BB	4	Program Once SACCH-2 Field (index = 0x0B)
0x3BC – 0x3BF	4	Program Once SACCL-2 Field (index = 0x0B)
0x3C0 – 0x3FF	64	Program Once ID Field (index = 0x00 - 0x07)

29.3.2.1 Program Once field

The Program Once field in the program flash 0 IFR provides 96 bytes of user data storage separate from the program flash 0 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 0 IFR is accessed in 8 byte records using the Read Once and Program Once commands (see Read Once Command and Program Once command).

29.3.3 Data flash 0 IFR map

The following only applies to devices with FlexNVM.

The data flash 0 IFR is a 1 Kbyte nonvolatile information memory that can be read and erased, but the user has limited program capabilities in the data flash 0 IFR (see the Program Partition command in Program Partition command, the Erase All Blocks command in Erase All Blocks Command, and the Read Resource command in Read Resource Command). The contents of the data flash 0 IFR are summarized in the following table and further described in the subsequent paragraphs.

The data flash 0 IFR is located within the data flash 0 memory block.

Address Range	Size (Bytes)	Field Description
0x00 – 0x3FB, 0x3FE – 0x3FF	1022	Reserved
0x3FD	1	EEPROM Data Set Size
0x3FC	1	FlexNVM Partition Code

29.3.3.1 EEPROM Data Set Size

The EEPROM data set size byte in the data flash 0 IFR supplies information which determines the amount of FlexRAM used in each of the available EEPROM subsystems. To program the EEERST, EEESIZE value, see the Program Partition command described in Program Partition command.

Table 29-1. EEPROM Data Set Size

Data flash IFR: 0x03FD							
7	6	5	4	3	2	1	0
1	EEERST	EEESPLIT		EEESIZE			
	= Unimplemented or Reserved						

Table 29-2. EEPROM Data Set Size Field Description

Field	Description
7 Reserved	This read-only bitfield is reserved and must always be written as one.
6 EEERST	**EEPROM Load on Reset** — Determines whether the flash reset sequence takes time to load the FlexRAM with valid EEPROM data. '0' = FlexRAM is not loaded with valid EEPROM data during the flash reset sequence (see the Set FlexRAM Function command described in Set FlexRAM Function command to load the FlexRAM with valid EEPROM data) '1' = FlexRAM is loaded with valid EEPROM data during the flash reset sequence
5-4 EEESPLIT	This read-only bitfield is reserved and each bit will always read as one.
3-0 EEESIZE	**EEPROM Size** — Encoding of the total available FlexRAM for EEPROM use. **NOTE:** EEESIZE must be 0 bytes (1111b) when the FlexNVM partition code (FlexNVM partition code) is set to 'No EEPROM'. '0000' = Reserved '0001' = 8,192 Bytes '0010' = 4,096 Bytes '0011' = 2,048 Bytes '0100' = 1,024 Bytes '0101' = 512 Bytes '0110' = 256 Bytes '0111' = 128 Bytes '1000' = 64 Bytes '1001' = 32 Bytes '1010' = Reserved '1011' = Reserved '1100' = Reserved '1101' = Reserved '1110' = Reserved '1111' = 0 Bytes

29.3.3.2 FlexNVM partition code

The FlexNVM partition code byte in the data flash 0 IFR supplies a code which specifies how to split the FlexNVM block between data flash memory and EEPROM backup memory supporting EEPROM functions. To program the DEPART value, see the Program Partition command described in Program Partition command.

Table 29-3. FlexNVM partition code

Data Flash IFR: 0x03FC							
7	6	5	4	3	2	1	0
1	1	1	1	DEPART			
	= Unimplemented or Reserved						

Table 29-4. FlexNVM partition code field description

Field	Description
7-4 Reserved	This read-only bitfield is reserved and must always be written as one.
3-0 DEPART	**FlexNVM Partition Code** — Encoding of the data flash / EEPROM backup split within the FlexNVM memory block. FlexNVM memory not partitioned for data flash is used to store EEPROM records.

DEPART	Data flash (KByte)	EEPROM backup (KByte)
0000	256	0
0001	Reserved	Reserved
0010	Reserved	Reserved
0011	224	32
0100	192	64
0101	128	128
0110	0	256
0111	Reserved	Reserved
1000	0	256
1001	Reserved	Reserved
1010	Reserved	Reserved
1011	32	224
1100	64	192
1101	128	128
1110	256	0
1111	256	0

29.3.4 Register descriptions

The FTFE 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).

FTFE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0000	Flash Status Register (FTFE_FSTAT)	8	R/W	00h	29.3.4.1/ 624
4002_0001	Flash Configuration Register (FTFE_FCNFG)	8	R/W	00h	29.3.4.2/ 625
4002_0002	Flash Security Register (FTFE_FSEC)	8	R	Undefined	29.3.4.3/ 627
4002_0003	Flash Option Register (FTFE_FOPT)	8	R	Undefined	29.3.4.4/ 629
4002_0004	Flash Common Command Object Registers (FTFE_FCCOB3)	8	R/W	00h	29.3.4.5/ 629
4002_0005	Flash Common Command Object Registers (FTFE_FCCOB2)	8	R/W	00h	29.3.4.5/ 629
4002_0006	Flash Common Command Object Registers (FTFE_FCCOB1)	8	R/W	00h	29.3.4.5/ 629
4002_0007	Flash Common Command Object Registers (FTFE_FCCOB0)	8	R/W	00h	29.3.4.5/ 629
4002_0008	Flash Common Command Object Registers (FTFE_FCCOB7)	8	R/W	00h	29.3.4.5/ 629
4002_0009	Flash Common Command Object Registers (FTFE_FCCOB6)	8	R/W	00h	29.3.4.5/ 629
4002_000A	Flash Common Command Object Registers (FTFE_FCCOB5)	8	R/W	00h	29.3.4.5/ 629
4002_000B	Flash Common Command Object Registers (FTFE_FCCOB4)	8	R/W	00h	29.3.4.5/ 629
4002_000C	Flash Common Command Object Registers (FTFE_FCCOBB)	8	R/W	00h	29.3.4.5/ 629
4002_000D	Flash Common Command Object Registers (FTFE_FCCOBA)	8	R/W	00h	29.3.4.5/ 629
4002_000E	Flash Common Command Object Registers (FTFE_FCCOB9)	8	R/W	00h	29.3.4.5/ 629
4002_000F	Flash Common Command Object Registers (FTFE_FCCOB8)	8	R/W	00h	29.3.4.5/ 629
4002_0010	Program Flash Protection Registers (FTFE_FPROT3)	8	R/W	Undefined	29.3.4.6/ 631
4002_0011	Program Flash Protection Registers (FTFE_FPROT2)	8	R/W	Undefined	29.3.4.6/ 631
4002_0012	Program Flash Protection Registers (FTFE_FPROT1)	8	R/W	Undefined	29.3.4.6/ 631

Table continues on the next page...

FTFE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0013	Program Flash Protection Registers (FTFE_FPROT0)	8	R/W	Undefined	29.3.4.6/ 631
4002_0016	EEPROM Protection Register (FTFE_FEPROT)	8	R/W	Undefined	29.3.4.7/ 632
4002_0017	Data Flash Protection Register (FTFE_FDPROT)	8	R/W	Undefined	29.3.4.8/ 633
4002_0018	Execute-only Access Registers (FTFE_XACCH3)	8	R	Undefined	29.3.4.9/ 634
4002_0019	Execute-only Access Registers (FTFE_XACCH2)	8	R	Undefined	29.3.4.9/ 634
4002_001A	Execute-only Access Registers (FTFE_XACCH1)	8	R	Undefined	29.3.4.9/ 634
4002_001B	Execute-only Access Registers (FTFE_XACCH0)	8	R	Undefined	29.3.4.9/ 634
4002_001C	Execute-only Access Registers (FTFE_XACCL3)	8	R	Undefined	29.3.4.9/ 634
4002_001D	Execute-only Access Registers (FTFE_XACCL2)	8	R	Undefined	29.3.4.9/ 634
4002_001E	Execute-only Access Registers (FTFE_XACCL1)	8	R	Undefined	29.3.4.9/ 634
4002_001F	Execute-only Access Registers (FTFE_XACCL0)	8	R	Undefined	29.3.4.9/ 634
4002_0020	Supervisor-only Access Registers (FTFE_SACCH3)	8	R	Undefined	29.3.4.10/ 635
4002_0021	Supervisor-only Access Registers (FTFE_SACCH2)	8	R	Undefined	29.3.4.10/ 635
4002_0022	Supervisor-only Access Registers (FTFE_SACCH1)	8	R	Undefined	29.3.4.10/ 635
4002_0023	Supervisor-only Access Registers (FTFE_SACCH0)	8	R	Undefined	29.3.4.10/ 635
4002_0024	Supervisor-only Access Registers (FTFE_SACCL3)	8	R	Undefined	29.3.4.10/ 635
4002_0025	Supervisor-only Access Registers (FTFE_SACCL2)	8	R	Undefined	29.3.4.10/ 635
4002_0026	Supervisor-only Access Registers (FTFE_SACCL1)	8	R	Undefined	29.3.4.10/ 635
4002_0027	Supervisor-only Access Registers (FTFE_SACCL0)	8	R	Undefined	29.3.4.10/ 635
4002_0028	Flash Access Segment Size Register (FTFE_FACSS)	8	R	Undefined	29.3.4.11/ 637
4002_002B	Flash Access Segment Number Register (FTFE_FACSN)	8	R	Undefined	29.3.4.12/ 637
4002_002E	Flash Error Status Register (FTFE_FERSTAT)	8	R/W	00h	29.3.4.13/ 638

Table continues on the next page...

FTFE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_002F	Flash Error Configuration Register (FTFE_FERCNFG)	8	R/W	00h	29.3.4.14/ 639

29.3.4.1 Flash Status Register (FTFE_FSTAT)

The FSTAT register reports the operational status of the FTFE 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 or writes to the FlexRAM (when EEERDY is set) 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

FTFE_FSTAT field descriptions

Field	Description
7 CCIF	Command Complete Interrupt Flag The CCIF flag indicates that a FTFE command or EEPROM file system operation 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. The CCIF flag is also cleared by a successful write to FlexRAM while enabled for EEPROM operations, and CCIF stays low until the EEPROM file system has created the associated EEPROM data record. The CCIF bit 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 FTFE command or EEPROM file system operation in progress 1 FTFE command or EEPROM file system operation has completed
6 RDCOLERR	FTFE Read Collision Error Flag The RDCOLERR error bit indicates that the MCU attempted a read from an FTFE resource that was being manipulated by an FTFE 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.

Table continues on the next page...

FTFE_FSTAT field descriptions (continued)

Field	Description
	0 No collision error detected 1 Collision error detected
5 ACCERR	Flash Access Error Flag The ACCERR error bit indicates an illegal access has occurred to an FTFE resource caused by a violation of the command write sequence or issuing an illegal FTFE 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 The FPVIOL error bit indicates an attempt was made to program or erase an address in a protected area of program flash or data flash memory during a command write sequence or a write was attempted to a protected area of the FlexRAM while enabled for EEPROM. 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 The MGSTAT0 status flag is set if an error is detected during execution of an FTFE command or during the flash reset sequence. As a status flag, this bit 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.

29.3.4.2 Flash Configuration Register (FTFE_FCNFG)

This register provides information on the current functional state of the FTFE module.

The erase control bits (ERSAREQ and ERSSUSP) have write restrictions. SWAP, PFLSH, RAMRDY, and EEERDY are read-only status bits. The reset values for the SWAP, PFLSH, RAMRDY, and EEERDY bits are determined during the reset sequence.

Address: 4002_0000h base + 1h offset = 4002_0001h

Bit	7	6	5	4	3	2	1	0
Read	CCIE	RDCOLLIE	ERSAREQ	ERSSUSP	SWAP	PFLSH	RAMRDY	EEERDY
Write								
Reset	0	0	0	0	0	0	0	0

FTFE_FCNFG field descriptions

Field	Description
7 CCIE	Command Complete Interrupt Enable The CCIE bit controls interrupt generation when an FTFE 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 The RDCOLLIE bit controls interrupt generation when an FTFE read collision error occurs. 0 Read collision error interrupt disabled 1 Read collision error interrupt enabled. An interrupt request is generated whenever an FTFE read collision error is detected (see the description of FSTAT[RDCOLERR]).
5 ERSAREQ	Erase All Request This bit 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. The ERSAREQ bit sets when an erase all request is triggered external to the FTFE and CCIF is set (no command is currently being executed). ERSAREQ is cleared by the FTFE when the operation completes. 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 The ERSSUSP bit 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 SWAP	Swap The SWAP flag indicates which half of the program flash space is located at relative address 0x0000. The state of the SWAP flag is set by the FTFE during the reset sequence. See for information on swap management. 0 For devices with FlexNVM: Program flash 0 block is located at relative address 0x0000 For devices with program flash only: Program flash 0 block is located at relative address 0x0000 1 For devices with program flash only: Program flash 1 block is located at relative address 0x0000
2 PFLSH	FTFE configuration 0 For devices with FlexNVM: FTFE configuration supports one program flash block and one FlexNVM block For devices with program flash only: Reserved 1 For devices with FlexNVM: Reserved For devices with program flash only: FTFE configuration supports two program flash blocks

Table continues on the next page...

FTFE_FCNFG field descriptions (continued)

Field	Description
1 RAMRDY	RAM Ready This flag indicates the current status of the FlexRAM/programming acceleration RAM. For devices with FlexNVM: The state of the RAMRDY flag is normally controlled by the Set FlexRAM Function command. During the reset sequence, the RAMRDY flag is cleared if the FlexNVM block is partitioned for EEPROM and will be set if the FlexNVM block is not partitioned for EEPROM . The RAMRDY flag is cleared if the Program Partition command is run to partition the FlexNVM block for EEPROM. The RAMRDY flag sets after completion of the Erase All Blocks command or execution of the erase-all operation triggered external to the FTFE. For devices without FlexNVM: This bit should always be set. 0 For devices with FlexNVM: FlexRAM is not available for traditional RAM access For devices without FlexNVM: Programming acceleration RAM is not available 1 For devices with FlexNVM: FlexRAM is available as traditional RAM only; writes to the FlexRAM do not trigger EEPROM operations For devices without FlexNVM: Programming acceleration RAM is available
0 EEERDY	For devices with FlexNVM: This flag indicates if the EEPROM backup data has been copied to the FlexRAM and is therefore available for read access. During the reset sequence, the EEERDY flag remains clear while CCIF=0 and only sets if the FlexNVM block is partitioned for EEPROM. For devices without FlexNVM: This bit is reserved and always has the value 0. 0 For devices with FlexNVM: FlexRAM is not available for EEPROM operation For devices without FlexNVM: See RAMRDY for availability of programming acceleration RAM 1 For devices with FlexNVM: FlexRAM is available for EEPROM operations where: • reads from the FlexRAM return data previously written to the FlexRAM in EEPROM mode and • writes launch an EEPROM operation to store the written data in the FlexRAM and EEPROM backup For devices without FlexNVM: Reserved

29.3.4.3 Flash Security Register (FTFE_FSEC)

This read-only register holds all bits associated with the security of the MCU and FTFE 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.

FTFE_FSEC field descriptions

Field	Description
7–6 KEYEN	Backdoor Key Security Enable These bits enable and disable backdoor key access to the FTFE module. 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 Bits Enables and disables mass erase capability of the FTFE module. When the SEC field 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	Factory Security Level Access Code These bits enable or disable access to the flash memory contents during returned part failure analysis at NXP. When SEC is secure and FSLACC is denied, access to the program flash contents is denied and any failure analysis performed by NXP 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), NXP factory testing has visibility of the current flash contents. The state of the FSLACC bits is only relevant when the SEC bits are set to secure. When the SEC field is set to unsecure, the FSLACC setting does not matter. 00 Factory access granted 01 Factory access denied 10 Factory access denied 11 Factory access granted
SEC	Flash Security These bits define the security state of the MCU. In the secure state, the MCU limits access to FTFE module resources. The limitations are defined per device and are detailed in the Chip Configuration details. If the FTFE module is unsecured using backdoor key access, the SEC bits are 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 FTFE is unsecure.) 11 MCU security status is secure

29.3.4.4 Flash Option Register (FTFE_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.

FTFE_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.

29.3.4.5 Flash Common Command Object Registers (FTFE_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

FTFE_FCCOB*n* field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>CCOBn</td><td>
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.

The following table shows a generic FTFE 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 FTFE command, typically an address and/or data values.

NOTE: 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.

<table>
<tr><th>FCCOB Number[1]</th><th>Typical Command Parameter Contents [7:0]</th></tr>
<tr><td>0</td><td>FCMD (a code that defines the FTFE 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>

FCCOB Endianness and Multi-Byte Access:

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).
</td></tr>
</table>

1. Refers to FCCOB register name, not register address

29.3.4.6 Program Flash Protection Registers (FTFE_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 FTFE command. Unprotected regions can be changed by program and erase operations.

The four FPROT registers allow up to 32 protectable regions of equal memory size.

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.

FTFE_FPROT*n* field descriptions

Field	Description
PROT	Program Flash Region Protect

FTFE_FPROT*n* field descriptions (continued)

Field	Description
	Each program flash region can be protected from program and erase operations by setting the associated PROT bit to the protected state. 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. **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. 0 Program flash region is protected. 1 Program flash region is not protected

29.3.4.7 EEPROM Protection Register (FTFE_FEPROT)

For devices with FlexNVM: The FEPROT register defines which EEPROM regions of the FlexRAM are protected against program and erase operations. Protected EEPROM regions cannot have their content changed by writing to it. Unprotected regions can be changed by writing to the FlexRAM.

For devices with program flash only: This register is reserved and not used.

Address: 4002_0000h base + 16h offset = 4002_0016h

Bit	7	6	5	4	3	2	1	0
Read Write	EPROT							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFE_FEPROT field descriptions

Field	Description
EPROT	EEPROM Region Protect For devices with program flash only: Reserved For devices with FlexNVM: Individual EEPROM regions can be protected from alteration by setting the associated EPROT bit to the protected state. The EPROT bits are not used when the FlexNVM Partition Code is set to data flash only. When the FlexNVM Partition Code is set to data flash and EEPROM or EEPROM only, each EPROT bit covers one-eighth of the configured EEPROM data (see the EEPROM Data Set Size parameter description).

FTFE_FEPROT field descriptions (continued)

Field	Description
	The protection can only be increased. This means 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 FEPROT bits with 1-to-0 transitions are accepted while all bits with 0-to-1 transitions are ignored. **Restriction:** Never write to the FEPROT register while a command is running (CCIF=0). **Reset:** During the reset sequence, the FEPROT register is loaded with the contents of the EEPROM protection byte in the Flash Configuration Field located in program flash. The flash basis for the reset values is signified by X in the register diagram. To change the EEPROM protection that will be loaded during the reset sequence, the sector of program flash that contains the Flash Configuration Field must be unprotected; then the EEPROM protection byte must be erased and reprogrammed. Trying to alter data by writing to any protected area in the EEPROM results in a protection violation error and sets the FSTAT[FPVIOL] bit. 0 For devices with program flash only: Reserved For devices with FlexNVM: EEPROM region is protected 1 For devices with program flash only: Reserved For devices with FlexNVM: EEPROM region is not protected

29.3.4.8 Data Flash Protection Register (FTFE_FDPROT)

The FDPROT register defines which data flash regions are protected against 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 FTFE command. Unprotected regions can be changed by both program and erase operations.

Address: 4002_0000h base + 17h offset = 4002_0017h

Bit	7	6	5	4	3	2	1	0
Read Write	DPROT							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFE_FDPROT field descriptions

Field	Description
DPROT	Data Flash Region Protect Individual data flash regions can be protected from program and erase operations by setting the associated DPROT bit to the protected state. Each DPROT bit protects one-eighth of the partitioned data flash memory space. The granularity of data flash protection cannot be less than the data flash sector

FTFE_FDPROT field descriptions (continued)

Field	Description
	size. If an unused DPROT bit is set to the protected state, the Erase all Blocks command does not execute and sets the FSTAT[FPVIOL] 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 FDPROT bits with 1-to-0 transitions are accepted while all bits with 0-to-1 transitions are ignored. **Restriction:** The user must never write to the FDPROT register while a command is running (CCIF=0). **Reset:** During the reset sequence, the FDPROT register is loaded with the contents of the data flash protection byte in the Flash Configuration Field located in program flash memory. The flash basis for the reset values is signified by X in the register diagram. To change the data flash protection that will be loaded during the reset sequence, unprotect the sector of program flash that contains the Flash Configuration Field. Then, erase and reprogram the data flash protection byte. Trying to alter data with the program and erase commands in any protected area in the data flash memory results in a protection violation error and sets the FSTAT[FPVIOL] bit. A block erase of any data flash memory block (see the Erase Flash Block command description) is not possible if the data flash block contains any protected region or if the FlexNVM memory has been partitioned for EEPROM. 0 Data Flash region is protected 1 Data Flash region is not protected

29.3.4.9 Execute-only Access Registers (FTFE_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	0x03A3	0x03AB
XACCH1	0x03A2	0x03AA
XACCH2	0x03A1	0x03A9
XACCH3	0x03A0	0x03A8
XACCL0	0x03A7	0x03AF
XACCL1	0x03A6	0x03AE
XACCL2	0x03A5	0x03AD
XACCL3	0x03A4	0x03AC

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.

FTFE_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

29.3.4.10 Supervisor-only Access Registers (FTFE_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]

Table continues on the next page...

Supervisor-only access register	Program flash supervisor-only access bits
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	0x03B3	0x03BB
SACCH1	0x03B2	0x03BA
SACCH2	0x03B1	0x03B9
SACCH3	0x03B0	0x03B8
SACCL0	0x03B7	0x03BF
SACCL1	0x03B6	0x03BE
SACCL2	0x03B5	0x03BD
SACCL3	0x03B4	0x03BC

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.

FTFE_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

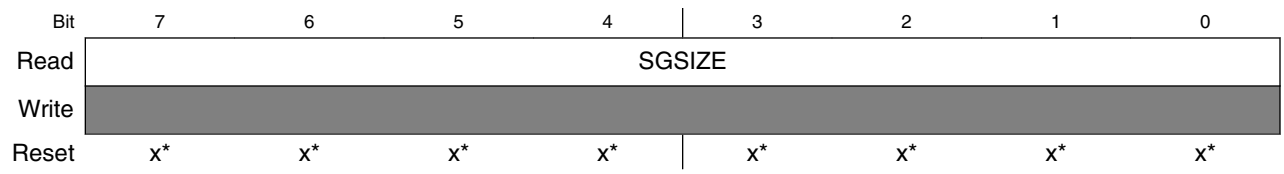
29.3.4.11 Flash Access Segment Size Register (FTFE_FACSS)

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



* Notes:
- x = Undefined at reset.

FTFE_FACSS field descriptions

Field	Description
SGSIZE	Segment Size The segment size is a fixed value based on the available program flash size divided by NUMSG.

Program Flash Size	Segment Size	Segment Size Encoding
256 KBytes	4 KBytes	0x4
512 KBytes	8 KBytes	0x5
768 KBytes	16 KBytes	0x6
1 MByte	16 KBytes	0x6
1.5 MBytes	32 KBytes	0x7
2 MBytes	32 KBytes	0x7

29.3.4.12 Flash Access Segment Number Register (FTFE_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.

FTFE_FACSN field descriptions

Field	Description
NUMSG	Number of Segments Indicator The NUMSG field indicates the number of equal-sized segments in the program flash. 0x30 Program flash memory is divided into 48 segments (768 Kbytes, 1.5 Mbytes) 0x40 Program flash memory is divided into 64 segments (256 Kbytes, 512 Kbytes, 1 Mbyte, 2 Mbytes)

29.3.4.13 Flash Error Status Register (FTFE_FERSTAT)

This register reports the detection of uncorrected ECC errors during read access to the FTFE module.

The DFDIF flag is readable and writable. The unassigned bits read 0 and are not writable.

Address: 4002_0000h base + 2Eh offset = 4002_002Eh

Bit	7	6	5	4	3	2	1	0
Read	0						DFDIF	0
Write							w1c	
Reset	0	0	0	0	0	0	0	0

FTFE_FERSTAT field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DFDIF	Double Bit Fault Detect Interrupt Flag The DFDIF flag indicates an uncorrectable ECC fault was detected during a valid flash read access from the platform flash controller. The DFDIF flag is cleared by writing a 1 to it. Writing a 0 to DFDIF has no effect. 0 Double bit fault not detected during a valid flash read access from the platform flash controller 1 Double bit fault detected (or FERCNFG[FDFD] is set) during a valid flash read access from the platform flash controller
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

29.3.4.14 Flash Error Configuration Register (FTFE_FERCNFG)

This register enables the force and interrupt of uncorrected ECC errors detected during read access to the FTFE module.

The FDFD and DFDIE bits are readable and writable. The unassigned bits read 0 and are not writable.

Address: 4002_0000h base + 2Fh offset = 4002_002Fh

Bit	7	6	5	4	3	2	1	0
Read	0		FDFD	0			DFDIE	0
Write								
Reset	0	0	0	0	0	0	0	0

FTFE_FERCNFG field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 FDFD	Force Double Bit Fault Detect The FDFD bit enables the user to emulate the setting of the FERSTAT[DFDIF] flag to check the associated interrupt routine. The FDFD bit is cleared by writing a 0 to FDFD. 0 FERSTAT[DFDIF] sets only if a double bit fault is detected during read access from the platform flash controller 1 FERSTAT[DFDIF] sets during any valid flash read access from the platform flash controller. An interrupt request is generated if the DFDIE bit is set.
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DFDIE	Double Bit Fault Detect Interrupt Enable The DFDIE bit controls interrupt generation when an uncorrectable ECC fault is detected during a valid flash read access from the platform flash controller. 0 Double bit fault detect interrupt disabled 1 Double bit fault detect interrupt enabled. An interrupt request is generated whenever the FERSTAT[DFDIF] flag is set.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

29.4 Functional Description

The following sections describe functional details of the FTFE module.

29.4.1 Program flash memory swap

The user can configure the memory map of the program flash space such that either half of the program flash memory can exist at relative address 0x0000. This swap feature enables the lower half of the program flash space to be operational while the upper half is being updated for future use.

The Swap Control command handles swapping the two halves of program flash memory within the memory map. See Swap Control command (program flash only devices) for details.

29.4.2 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* — Four registers protect 32 regions of the program flash memory as shown in the following figure

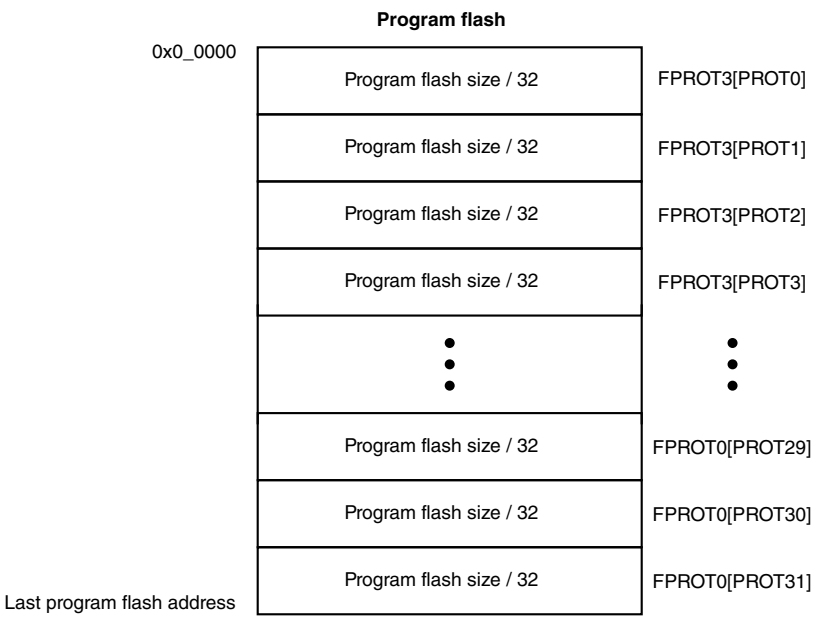


Figure 29-3. Program flash protection

- FDPROT —
  - For 2^n data flash sizes, protects eight regions of the data flash memory as shown in the following figure

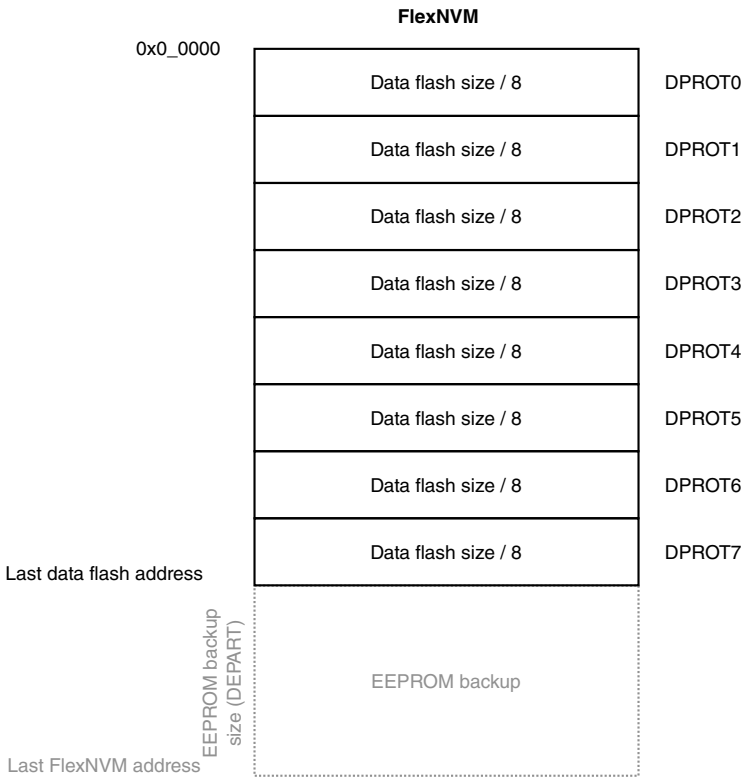


Figure 29-4. Data flash protection (2^n data flash sizes)

- FEPROT — Protects eight regions of the EEPROM memory as shown in the following figure

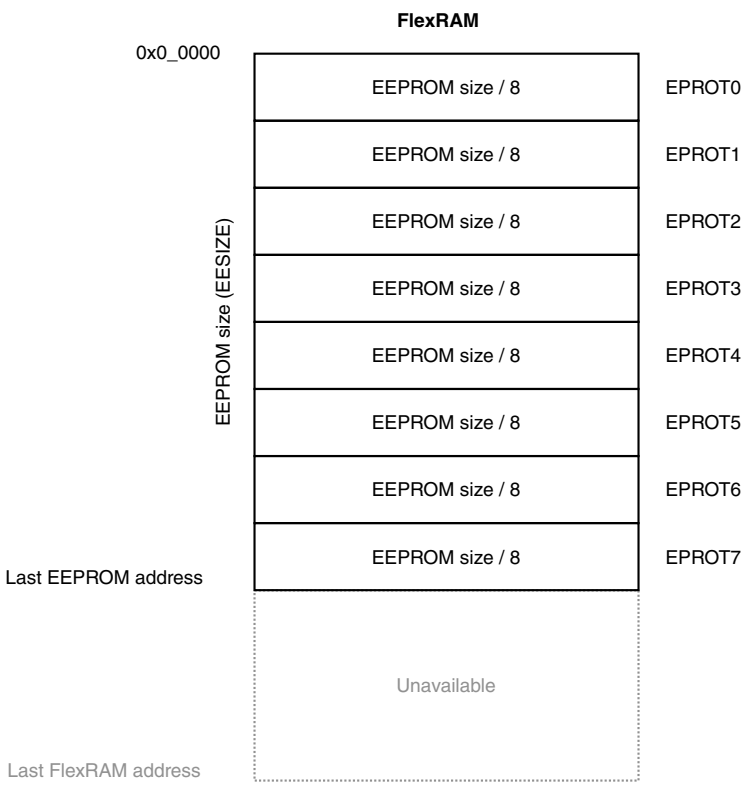


Figure 29-5. EEPROM protection

NOTE

Flash protection features are discussed further in AN4507: Using the Kinetis Security and Flash Protection Features . Some features described in the application note may not be available on this device.

29.4.3 Flash Access Protection

Individual segments within the program flash memory can be designated for restricted access. Specific flash commands (Program Check, Program Phrase, Erase Flash Block, Erase Flash Sector) monitor FXACC contents to protect flash memory but the FSACC contents do not impact flash command operation.

See AN5112: Using the Kinetis Flash Execute-Only Access Control Feature for further details.

Access is controlled by the following registers:

- FXACC —
  - eight registers control 64 segments of the program flash memory as shown in the following figure

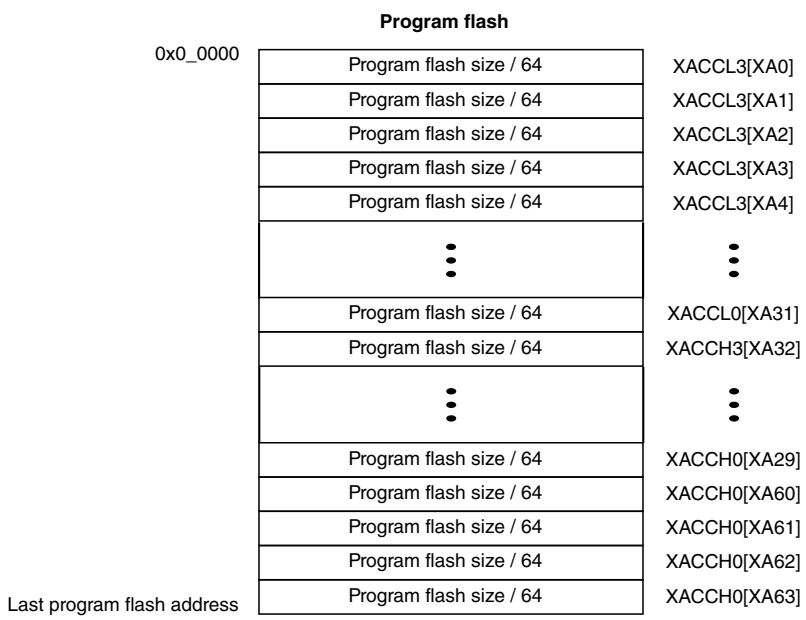


Figure 29-6. Program flash execute-only access control

- FSACC —
  - eight registers control 64 segments of the program flash memory as shown in the following figure

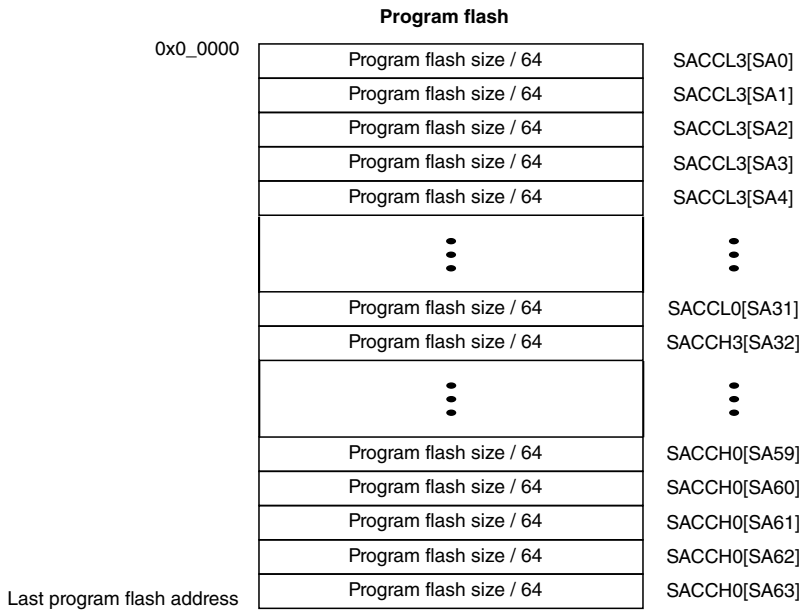


Figure 29-7. Program flash supervisor access control

29.4.3.1 FAC Application Tips

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.

29.4.4 FlexNVM Description

This section describes the FlexNVM memory. This section does not apply for devices that contain only program flash memory.

29.4.4.1 FlexNVM Block Partitioning for FlexRAM

The user can configure the FlexNVM block as either:

- Basic data flash,
- EEPROM flash records to support the built-in EEPROM feature, or
- A combination of both.

The user's FlexNVM configuration choice is specified using the Program Partition command described in Program Partition command.

CAUTION

While different partitions of the FlexNVM block are available, the intention is that a single partition choice is used throughout the entire lifetime of a given application. The FlexNVM partition code choices affect the endurance and data retention characteristics of the device.

29.4.4.2 EEPROM User Perspective

The EEPROM system is shown in the following figure.

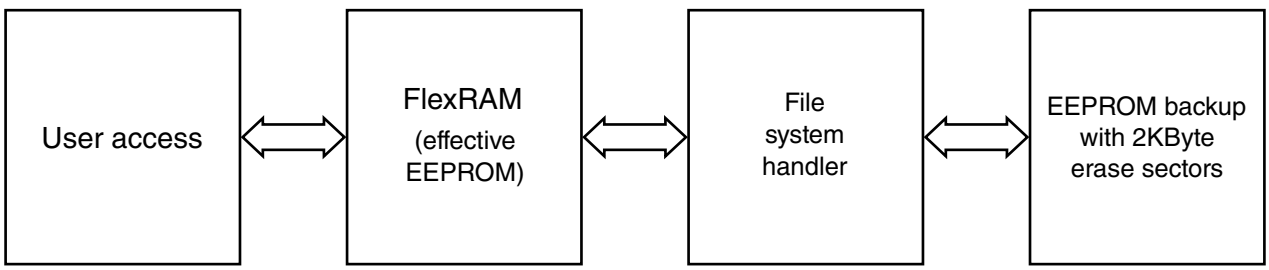


Figure 29-8. Top Level EEPROM Architecture

To handle varying customer requirements, the FlexRAM and FlexNVM blocks can be split into partitions as shown in the figure below.

1. **EEPROM partition** (EEESIZE) — The amount of FlexRAM used for EEPROM can be set from 0 Bytes (no EEPROM) to the maximum FlexRAM size (see Table 29-2). The remainder of the FlexRAM not used for EEPROM is not accessible while the FlexRAM is configured for EEPROM (see Set FlexRAM Function command). The EEPROM partition grows upward from the bottom of the FlexRAM address space.
2. **Data flash partition** (DEPART) — The amount of FlexNVM memory used for data flash can be programmed from 0 bytes (all of the FlexNVM block is available for EEPROM backup) to the maximum size of the FlexNVM block (see Table 29-4).
3. **FlexNVM EEPROM partition** — The amount of FlexNVM memory used for EEPROM backup, which is equal to the FlexNVM block size minus the data flash memory partition size. The EEPROM backup size must be at least 16 times the EEPROM partition size in FlexRAM.

The partition information (EEESIZE, DEPART) is stored in the data flash IFR and is programmed using the Program Partition command (see Program Partition command). Typically, the Program Partition command is executed only once in the lifetime of the device.

Data flash memory is useful for applications that need to quickly store large amounts of data or store data that is static. The EEPROM partition in FlexRAM is useful for storing smaller amounts of data that will be changed often.

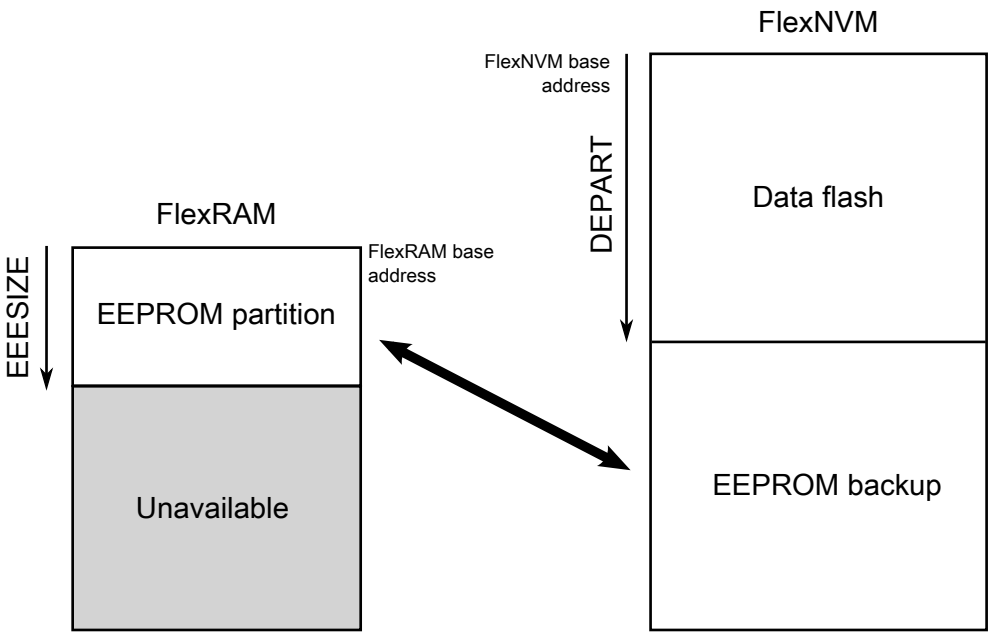


Figure 29-9. FlexRAM to FlexNVM Memory Mapping for EEPROM

29.4.4.3 EEPROM implementation overview

Out of reset with the FSTAT[CCIF] bit clear, the partition settings (EEESIZE, DEPART) are read from the data flash IFR and the EEPROM file system is initialized accordingly. The EEPROM file system locates all valid EEPROM data records in EEPROM backup and copies the newest data to FlexRAM. The FSTAT[CCIF] and FCNFG[EEERDY] bits are set after data from all valid EEPROM data records is copied to the FlexRAM. After the CCIF bit is set, the FlexRAM is available for read or write access.

When configured for EEPROM use, writes to an unprotected location in FlexRAM invokes the EEPROM file system to program a new EEPROM data record in the EEPROM backup memory in a round-robin fashion. As needed, the EEPROM file system identifies the EEPROM backup sector that is being erased for future use and partially erases that EEPROM backup sector. After a write to the FlexRAM, the FlexRAM is not accessible until the FSTAT[CCIF] bit is set. The FCNFG[EEERDY] bit will also be set. If enabled, the interrupt associated with the FSTAT[CCIF] bit can be used to determine when the FlexRAM is available for read or write access.

When configured for EEPROM use, attempts to write to the FlexRAM are ignored in VLP mode or while the radio is active .

After a sector in EEPROM backup is full of EEPROM data records, EEPROM data records from the sector holding the oldest data are gradually copied over to a previously-erased EEPROM backup sector. When the sector copy completes, the EEPROM backup sector holding the oldest data is tagged for erase.

29.4.4.4 Impact of radio activity on EEPROM operations

The RSIM module provides an early indicator to the flash module that the radio is going active and also indicates when the radio goes inactive. If the radio active indicator is asserted while an EEPROM operation is active, the operation will stall to reduce power consumption before the radio goes active. The EEPROM operation will resume after the radio active indicator negates. FSTAT[CCIF] and EEERDY remain clear during the stall to prevent disruption of the EEPROM operation while the data flash and FlexRAM remain unavailable for read or write operations.

The following figure shows how the EEPROM write operation stalls and resumes based on radio activity.

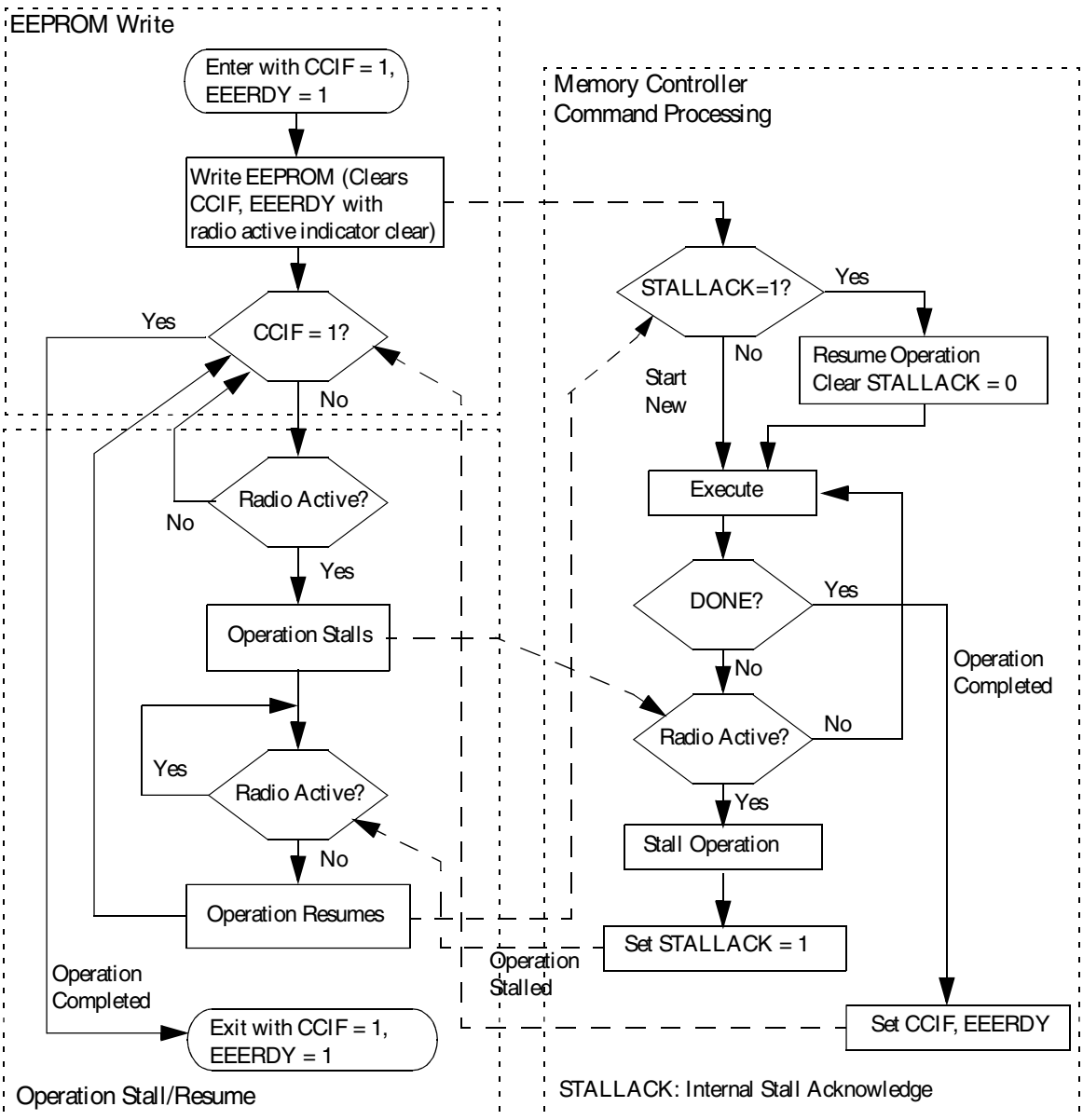


Figure 29-10. Radio Active Impact on EEPROM Write Operation

29.4.4.5 Write endurance to FlexRAM for EEPROM

When the FlexNVM partition code is not set to full data flash, the EEPROM data set size can be set to any of several non-zero values.

The bytes not assigned to data flash via the FlexNVM partition code are used by the FTFE to obtain an effective endurance increase for the EEPROM data. The built-in EEPROM record management system raises the number of program/erase cycles that can be attained prior to device wear-out by cycling the EEPROM data through a larger EEPROM NVM storage space.

While different partitions of the FlexNVM are available, the intention is that a single choice for the FlexNVM partition code and EEPROM data set size is used throughout the entire lifetime of a given application.

29.4.5 Interrupts

The FTFE module can generate interrupt requests to the MCU upon the occurrence of various FTFE events. These interrupt events and their associated status and control bits are shown in the following table.

Table 29-5. FTFE Interrupt Sources

FTFE Event	Readable Status Bit	Interrupt Enable Bit
FTFE Command Complete	FSTAT[CCIF]	FCNFG[CCIE]
FTFE Read Collision Error	FSTAT[RDCOLERR]	FCNFG[RDCOLLIE]
FTFE ECC Error Detection	FERSTAT[DFDIF]	FERCNFG[DFDIE]

Note

Vector addresses and their relative interrupt priority are determined at the MCU level.

29.4.6 Flash Operation in Low-Power Modes

29.4.6.1 Wait Mode

When the MCU enters wait mode, the FTFE module is not affected. The FTFE module can recover the MCU from wait via the command complete interrupt (see Interrupts).

29.4.6.2 Stop Mode

When the MCU requests stop mode, if an FTFE 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 FTFE command is running (CCIF = 0).

NOTE

While the MCU is in very-low-power modes (VLPR, VLPW, VLPS), the FTFE module does not accept flash commands.

29.4.7 Flash memory reads and ignored writes

The FTFE module requires only the flash address to execute a flash memory read. MCU read access is available to all flash memory.

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).

29.4.8 Read while write (RWW)

The following simultaneous accesses are allowed for devices with FlexNVM:

- The user may read from the program flash memory while commands (typically program and erase operations) are active in the data flash and FlexRAM memory space.
- The MCU can fetch instructions from program flash during both data flash program and erase operations and while EEPROM-backup is maintained by the EEPROM commands.
- Conversely, the user may read from data flash and FlexRAM while program and erase commands are executing on the program flash.
- When configured as traditional RAM, writes to the FlexRAM are allowed during data flash operations.

Simultaneous data flash operations and FlexRAM writes, when FlexRAM is used as EEPROM, are not possible.

The following simultaneous accesses are allowed for devices with program flash only:

- The user may read from one program flash memory block while commands are active in the other program flash memory block.

Simultaneous operations are further discussed in Allowed simultaneous flash operations.

29.4.9 Flash Program and Erase

All flash functions except read require the user to setup and launch an FTFE command through a series of peripheral bus writes. The user cannot initiate any further FTFE commands until notified that the current command has completed. The FTFE command structure and operation are detailed in FTFE Command Operations.

29.4.10 FTFE Command Operations

FTFE command operations are typically used to modify flash memory contents. The next sections describe:

- The command write sequence used to set FTFE command parameters and launch execution
- A description of all FTFE commands available

29.4.10.1 Command Write Sequence

FTFE commands are specified using a command write sequence illustrated in Figure 29-11. The FTFE 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 an FTFE command in VLP mode will be ignored. Attempts to launch a flash command while the radio active indicator is asserted are ignored.

29.4.10.1.1 Load the FCCOB Registers

The user must load the FCCOB registers with all parameters required by the desired FTFE command. The individual registers that make up the FCCOB data set can be written in any order.

29.4.10.1.2 Launch the Command by Clearing CCIF

Once all relevant command parameters have been loaded, the user launches the command by clearing the FSTAT[CCIF] bit by writing a '1' to it. The CCIF flag remains zero until the FTFE command completes.

The FSTAT register contains a blocking mechanism, which prevents a new command from launching (can't clear 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.

29.4.10.1.3 Command Execution and Error Reporting

The command processing has several steps:

1. The FTFE 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. 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, the 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 the FSTAT[CCIF] bit.

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 the FSTAT[MGSTAT0] bit. 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 FTFE sets the FSTAT[CCIF] bit signifying that the command has completed.

The flow for a generic command write sequence is illustrated in the following figure.

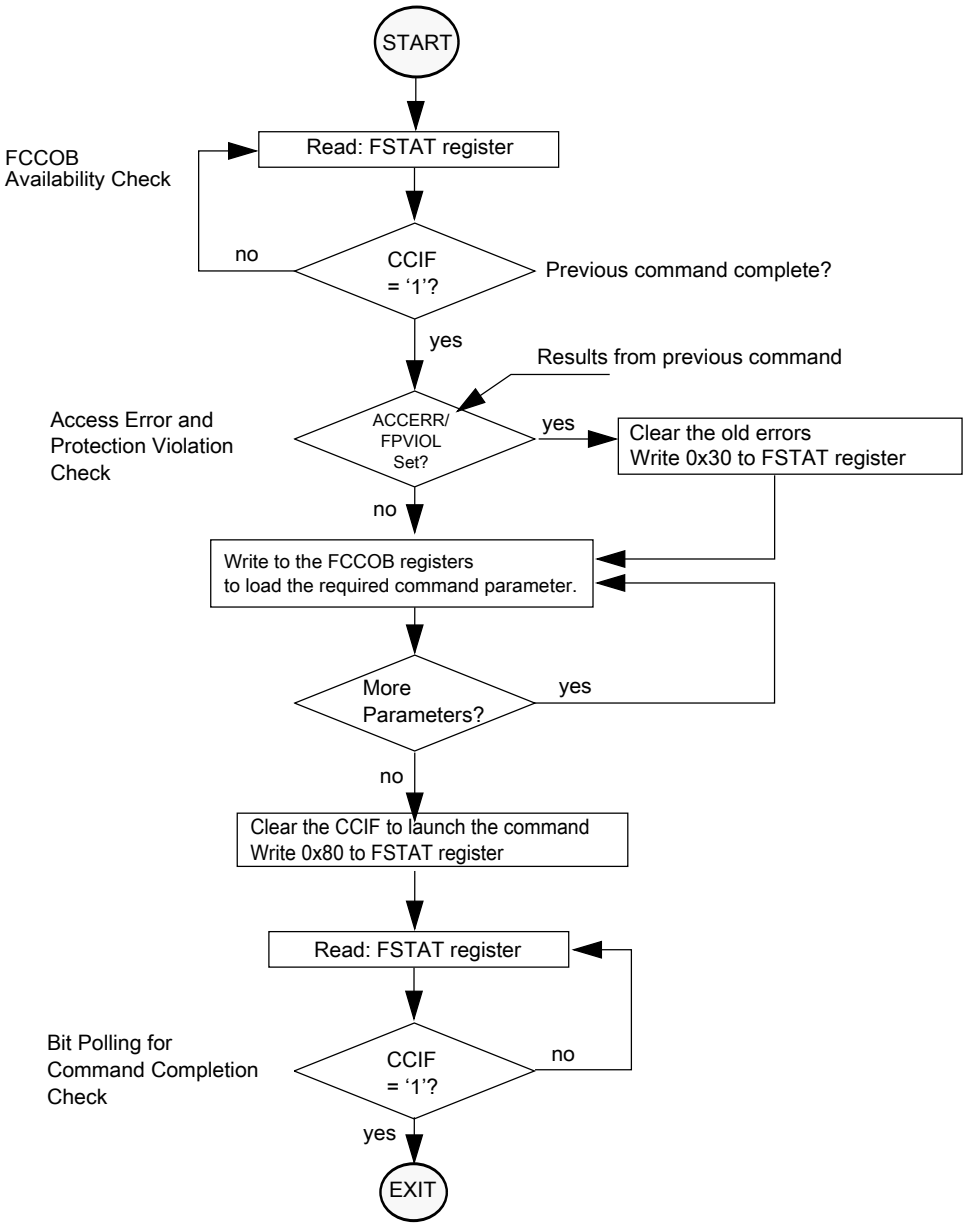


Figure 29-11. Generic Flash Command Write Sequence Flowchart

29.4.10.2 Flash commands

The following table summarizes the function of all flash commands. If any column is marked with an 'X', the flash command is relevant to that particular memory resource.

FCMD	Command	Program flash 0	Program flash 1 (Devices with only program flash)	Data flash (Devices with FlexNVM)	FlexRAM (Devices with FlexNVM)	Function
0x00	Read 1s Block	×	×	×		Verify that a program flash or data flash block is erased. FlexNVM block must not be partitioned for EEPROM.
0x01	Read 1s Section	×	×	×		Verify that a given number of program flash or data flash locations from a starting address are erased.
0x02	Program Check	×	×	×		Tests previously-programmed phrases at margin read levels.
0x03	Read Resource	IFR,ID	IFR	IFR		Read 8 bytes from program flash IFR, data flash IFR, or version ID.
0x07	Program Phrase	×	×	×		Program 8 bytes in a program flash block or a data flash block.
0x08	Erase Flash Block	×	×	×		Erase a program flash block or data flash block. An erase of any flash block is only possible when unprotected. FlexNVM block must not be partitioned for EEPROM.
0x09	Erase Flash Sector	×	×	×		Erase all bytes in a program flash or data flash sector.

Table continues on the next page...

FCMD	Command	Program flash 0	Program flash 1 (Devices with only program flash)	Data flash (Devices with FlexNVM)	FlexRAM (Devices with FlexNVM)	Function
0x0B	Program Section	×	×	×	×	Program data from the Section Program Buffer to a program flash or data flash block.
0x40	Read 1s All Blocks	×	×	×		Verify that all program flash, data flash blocks, EEPROM backup data records, and data flash IFR are erased then release MCU security.
0x41	Read Once	IFR				Read 8 bytes of a dedicated 64 byte field in the program flash 0 IFR.
0x43	Program Once	IFR				One-time program of 8 bytes of a dedicated 64-byte field in the program flash 0 IFR.
0x44	Erase All Blocks	×	×	×	×	Erase all program flash blocks, program flash swap IFR, data flash blocks, FlexRAM, EEPROM backup data records, and data flash IFR. Then, verify-erase and release MCU security. **NOTE**: An erase is only possible when

Table continues on the next page...

FCMD	Command	Program flash 0	Program flash 1 (Devices with only program flash)	Data flash (Devices with FlexNVM)	FlexRAM (Devices with FlexNVM)	Function
						all memory locations are unprotected.
0x45	Verify Backdoor Access Key	×	x			Release MCU security after comparing a set of user-supplied security keys to those stored in the program flash.
0x46	Swap Control	x	x			Handles swap-related activities.
0x49	Erase All Blocks Unsecure	×	×	×	×	Erase all program flash blocks, program flash swap IFR, data flash blocks, FlexRAM, EEPROM backup data records, and data flash IFR. Then, verify-erase, program the security byte to the unsecure state, and release MCU security.
0x4A	Read 1s All Execute-only Segments	x	x			Verify that all program flash execute-only (XA) segments are erased then release flash access control.
0x4B	Erase All Execute-only Segments	x	x			Erase all program flash execute-only (XA) segments then release flash access control.
0x80	Program Partition			IFR, ×	×	Program the FlexNVM Partition Code and EEPROM

Table continues on the next page...

FCMD	Command	Program flash 0	Program flash 1 (Devices with only program flash)	Data flash (Devices with FlexNVM)	FlexRAM (Devices with FlexNVM)	Function
						Data Set Size into the data flash IFR. format all EEPROM backup data sectors allocated for EEPROM, initialize the FlexRAM.
0x81	Set FlexRAM Function			×	×	Switches FlexRAM function between RAM and EEPROM. When switching to EEPROM, FlexNVM is not available while valid data records are being copied from EEPROM backup to FlexRAM.

29.4.10.3 Allowed simultaneous flash operations

Only the operations marked 'OK' in the following table are permitted to run simultaneously on the program flash, data flash, and FlexRAM memories. Some operations cannot be executed simultaneously because certain hardware resources are shared by the memories. The priority has been placed on permitting program flash reads while program and erase operations execute on the FlexNVM and FlexRAM. This provides read (program flash) while write (FlexNVM, FlexRAM) functionality.

For devices containing FlexNVM:

Table 29-6. Allowed Simultaneous Memory Operations

		Program flash			Data flash			FlexRAM		
		Read	Program Phrase	Erase Flash Sector[1]	Read	Program Phrase	Erase Flash Sector[1]	Read	E-Write[2]	R-Write[3]
Program flash	Read					OK	OK		OK	
	Program Phrase				OK			OK		OK
	Erase Flash Sector[1]				OK			OK		OK
Data flash	Read		OK	OK						
	Program Phrase	OK						OK		OK
	Erase Flash Sector[1]	OK						OK		OK
FlexRAM	Read		OK	OK		OK	OK			
	E-Write[2]	OK								
	R-Write[3]		OK	OK		OK	OK			

1. Also applies to Erase Flash Block
2. When FlexRAM configured for EEPROM (EEERDY=1).
3. When FlexRAM configured as traditional RAM (RAMRDY=1); single cycle operation.

For devices containing program flash only:

Table 29-7. Allowed Simultaneous Memory Operations

		Program flash X[1]			
		Read	Program Phrase	Erase Flash Sector	Erase Flash Block
Program flash Y[1]	Read		OK	OK	OK
	Program Phrase	OK			
	Erase Flash Sector	OK			
	Erase Flash Block	OK			

1. P-Flash X refers to any of the P-Flash blocks (0, 1) and P-Flash Y refers to any of the P-Flash blocks (0, 1), but not the same block. Thus, it is possible to read from any of the blocks while programming or erasing another.

29.4.11 Margin Read Commands

The Read-1s commands (Read 1s All Blocks, Read 1s Block, Read 1s Section, Read 1s All Execute-only Segments) 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 and data 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. All simple (uncommanded) flash array reads to the MCU always 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.

29.4.12 Flash command descriptions

This section describes all flash commands that can be launched by a command write sequence. The FTFE 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 the ACCERR and FPVIOL bits in the FSTAT register 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 FTFE is running a command (CCIF = 0) on that same block. The FTFE may return invalid data to the MCU with the collision error flag (FSTAT[RDCOLERR]) set.

When required by the command, address bit 23 selects between program flash memory (=0) and data flash memory (=1).

CAUTION

Flash data must be in the erased state before being programmed. Cumulative programming of bits (adding more zeros) is not allowed.

29.4.12.1 Read 1s Block command

The Read 1s Block command checks to see if an entire program flash or data flash block has been erased to the specified margin level. The FCCOB flash address bits determine which block is erase-verified.

Table 29-8. 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 64-bit aligned (Flash address [2:0] = 000).

After clearing CCIF to launch the Read 1s Block command, the FTFE sets the read margin for 1s according to Table 29-9 and then reads all locations within the selected program flash or data flash block.

When the data flash is targeted, DEPART must be set for no EEPROM, else the Read 1s Block command aborts setting the FSTAT[ACCERR] bit. If the FTFE fails to read all 1s (i.e. the flash block is not fully erased), the FSTAT[MGSTAT0] bit is set. The CCIF flag sets after the Read 1s Block operation has completed.

CAUTION

The Read 1s Block operation will not react to the early indicator for radio activity. Therefore, the Read 1s Block command must not be launched if there is a concern about inadequate power available to support both radio activity and the Read 1s Block operation.

Table 29-9. 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 29-10. 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]
Data flash is selected and the address is out of data flash range	FSTAT[ACCERR]
Data flash is selected with EEPROM enabled	FSTAT[ACCERR]
Flash address is not 64-bit aligned	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

29.4.12.2 Read 1s Section command

The Read 1s Section command checks if a section of program flash or data flash memory is erased to the specified read margin level. The Read 1s Section command defines the starting address and the number of phrases to be verified.

Table 29-11. Read 1s Section Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x01 (RD1SEC)
1	Flash address [23:16] of the first phrase to be verified

Table continues on the next page...

Table 29-11. Read 1s Section Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents [7:0]
2	Flash address [15:8] of the first phrase to be verified
3	Flash address [7:0][1] of the first phrase to be verified
4	Number of phrases to be verified [15:8]
5	Number of phrases to be verified [7:0]
6	Read-1 Margin Choice

1. Must be 64-bit aligned (Flash address [2:0] = 000).

Upon clearing CCIF to launch the Read 1s Section command, the FTFE sets the read margin for 1s according to Table 29-12 and then reads all locations within the specified section of flash memory.

If the FTFE fails to read all 1s (i.e. the flash section is not erased), the FSTAT(MGSTAT0) bit is set. The CCIF flag sets after the Read 1s Section operation completes.

CAUTION

The Read 1s Section operation will not react to the early indicator for radio activity. Therefore, the Read 1s Section command must not be launched if there is a concern about inadequate power available to support both radio activity and the Read 1s Section operation if the number of phrases to be verified is greater than 10,240.

Table 29-12. 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 29-13. 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 64-bit aligned	FSTAT[ACCERR]
The requested section crosses a flash block boundary	FSTAT[ACCERR]
The requested number of phrases is zero	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

29.4.12.3 Program Check command

The Program Check command tests a previously programmed program flash or data flash longword to see if it reads correctly at the specified margin level.

Table 29-14. 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 FTFE sets the read margin for 1s based on the provided margin choice according to Table 29-15. The Program Check operation then 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, the MGSTAT0 bit is set.

The FTFE will then set the read margin for 0s based on the provided margin choice.The Program Check operation will then read the specified longword and compare the actual read data to the expected data provided by the FCCOB. If the comparison at margin-0 fails, the MGSTAT0 bit will be set. The CCIF flag will set after the Program Check operation has completed.

The starting address must be longword aligned (the lowest two bits of the byte address must be 00):

- Byte 0 data is expected at the supplied 32-bit aligned address,
- Byte 1 data is expected at byte address specified + 0b01,
- Byte 2 data is expected at byte address specified + 0b10, and
- Byte 3 data is expected at byte address specified + 0b11.

NOTE

See the description of margin reads, Margin Read Commands

Table 29-15. 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 29-16. 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, Erase All Blocks Unsecure 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]

29.4.12.4 Read Resource Command

The Read Resource command is provided for the user to read data from special-purpose memory resources located within the Flash module. The special-purpose memory resources available include program flash IFR, data flash IFR space, and the Version ID field. The Version ID field contains an 8 byte code that indicates a specific FTFE implementation.

Table 29-17. 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]
4	Resource select code (see Table 29-18)
Returned values	
4	Read Data [64:56]
5	Read Data [55:48]
6	Read Data [47:40]
7	Read Data [39:32]
8	Read Data [31:24]
9	Read Data [23:16]
A	Read Data [15:8]

Table continues on the next page...

Table 29-17. Read Resource Command FCCOB Requirements (continued)

FCCOB Number	FCCOB contents [7:0]
B	Read Data [7:0]

1. Must be 64-bit aligned (Flash address [2:0] = 000).

Table 29-18. Read Resource Select Codes

Resource Select Code	Description	Resource Size	Local Address Range
0x00	Program Flash 0 IFR	1024 Bytes	0x00_0000 - 0x00_03FF
0x00	Program Flash Swap IFR	1024 Bytes	0x04_0000 - 0x04_03FF
0x00	Data Flash 0 IFR	1024 Bytes	0x80_0000 - 0x80_03FF
0x01	Version ID	8 Bytes	0x00_0008 - 0x00_000F

After clearing CCIF to launch the Read Resource command, eight consecutive bytes are read from the selected resource at the provided relative address and stored in the FCCOB register. The CCIF flag will set after the Read Resource operation has completed. 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 29-19. 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 64-bit aligned	FSTAT[ACCERR]

29.4.12.5 Program Phrase command

The Program Phrase command programs eight previously-erased bytes in the program flash memory or in the data 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 29-20. Program Phrase Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x07 (PGM8)
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
8	Byte 4 program value
9	Byte 5 program value
A	Byte 6 program value
B	Byte 7 program value

1. Must be 64-bit aligned (Flash address [2:0] = 000)

Upon clearing CCIF to launch the Program Phrase command, the FTFE programs the data bytes into the flash using the supplied address. The protection status is always checked. If the swap system is enabled, the double-phrase containing the swap indicator address in each half of the program flash space is implicitly protected from programming. The targeted flash locations must be currently unprotected (see the description of the FPROT registers) to permit execution of the Program Phrase 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 MGSTAT0. The CCIF flag is set after the Program Phrase operation completes.

The starting address must be 64-bit aligned (flash address [2:0] = 000):

- Byte 0 data is written to the starting address ('start'),
- Byte 1 data is programmed to byte address start+0b01,
- Byte 2 data is programmed to byte address start+0b10, and
- Byte 3 data is programmed to byte address start+0b11, etc.

Table 29-21. Program Phrase 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 64-bit aligned	FSTAT[ACCERR]
Flash address points to a protected area	FSTAT[FPVIOL]

Table continues on the next page...

Table 29-21. Program Phrase Command Error Handling (continued)

Error Condition	Error Bit
Flash address is located in an XA controlled segment and the Erase All Blocks, Erase All Blocks Unsecure 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]

29.4.12.6 Erase Flash Block Command

The Erase Flash Block operation erases all addresses in a single program flash or data flash block.

Table 29-22. 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 64-bit aligned (Flash address [2:0] = 000).

Upon clearing CCIF to launch the Erase Flash Block command, the FTFE erases the main array of the selected flash block and verifies that it is erased. When the data flash is targeted, DEPART must be set for no EEPROM (see Table 29-4) else the Erase Flash Block command aborts setting the FSTAT[ACCERR] bit. 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 program flash protection (FPROT) registers and the data flash protection (FDPROT) registers). If the swap system is enabled, the swap indicator address is implicitly protected from block erase unless the swap system is in the UPDATE or UPDATE-ERASED state and the program flash block being erased is the non-active block that contains the swap indicator address. If the erase verify fails, the MGSTAT0 bit in FSTAT is set. The CCIF flag will set after the Erase Flash Block operation has completed.

Table 29-23. 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]
Data flash is selected and the address is out of data flash range	FSTAT[ACCERR]
Data flash is selected with EEPROM enabled	FSTAT[ACCERR]

Table continues on the next page...

Table 29-23. Erase Flash Block Command Error Handling (continued)

Error Condition	Error Bit
Flash address is not 64-bit aligned	FSTAT[ACCERR]
Any area of the selected flash block is protected	FSTAT[FPVIOL]
The selected program flash block contains an XA controlled segment and the Erase All Blocks, Erase All Blocks Unsecure 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.

29.4.12.6.1 Impact of radio activity on Erase Flash Block operation

The RSIM module provides an early indicator to the flash module that the radio is going active and also indicates when the radio goes inactive. If the radio active indicator is asserted while the Erase Flash Block operation is active, the operation will stall to reduce power consumption before the radio goes active. The Erase Flash Block operation will resume after the radio active indicator negates. FSTAT[CCIF] remains clear during the stall to prevent disruption of the Erase Flash Block operation while the block remains unavailable for read operations.

The following figure shows how the Erase Flash Block operation stalls and resumes based on radio activity.

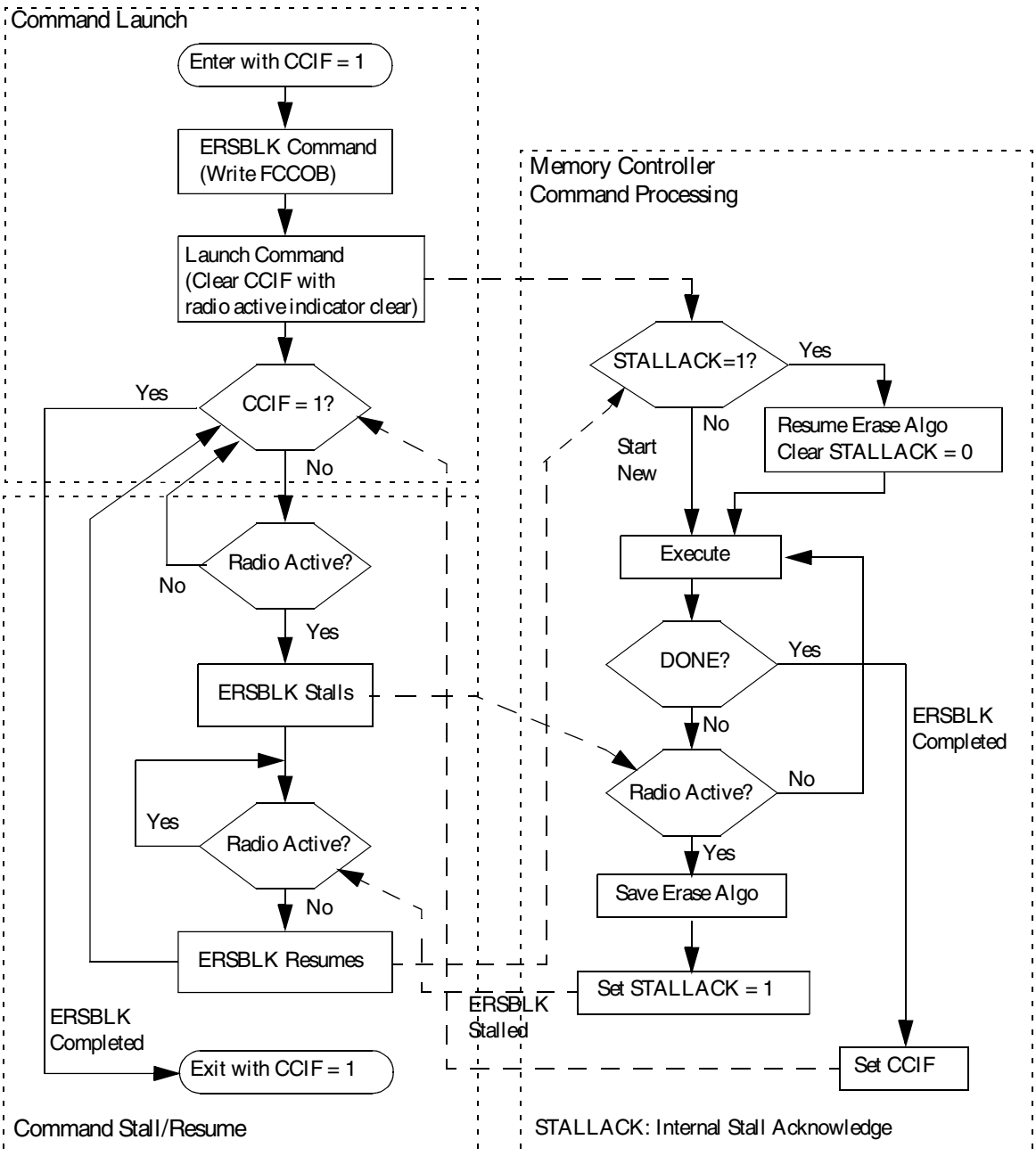


Figure 29-12. Radio Active Impact on Erase Flash Block Operation

29.4.12.7 Erase Flash Sector command

The Erase Flash Sector operation erases all addresses in a flash sector.

Table 29-24. Erase Flash Sector Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x09 (ERSSCR)

Table continues on the next page...

Table 29-24. Erase Flash Sector Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents [7:0]
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][1] in the flash sector to be erased

1. Must be 64-bit aligned (Flash address [2:0] = 000).

After clearing CCIF to launch the Erase Flash Sector command, the FTFE erases the selected program flash or data 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 swap system is enabled, the swap indicator address in each program flash block is implicitly protected from sector erase unless the swap system is in the UPDATE or UPDATE-ERASED state and the program flash sector containing the swap indicator address being erased is in the non-active block. 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 29-13).

Table 29-25. 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 64-bit aligned	FSTAT[ACCERR]
The selected program flash or data flash sector is protected	FSTAT[FPVIOL]
The selected program flash sector is located in an XA controlled segment and the Erase All Blocks, Erase All Blocks Unsecure 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.

29.4.12.7.1 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 samples the state of the ERSSUSP bit at convenient points. If the FTFE detects that the ERSSUSP bit is set, the Erase Flash Sector operation is suspended and the FTFE 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 FTFE detects that a suspend request has been made, the FTFE clears the ERSSUSP bit prior to setting CCIF. When an Erase Flash Sector operation has been successfully suspended, the FTFE 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 FTFE has acknowledged it.

29.4.12.7.2 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 FTFE 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.

29.4.12.7.3 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 FTFE starts the new command using the new FCCOB contents.

While FCNFG[ERSSUSP] is set, a write to the FlexRAM while FCNFG[EEERDY] is set clears ERSSUSP and aborts the suspended operation. The FlexRAM write operation is executed by the FTFE.

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.

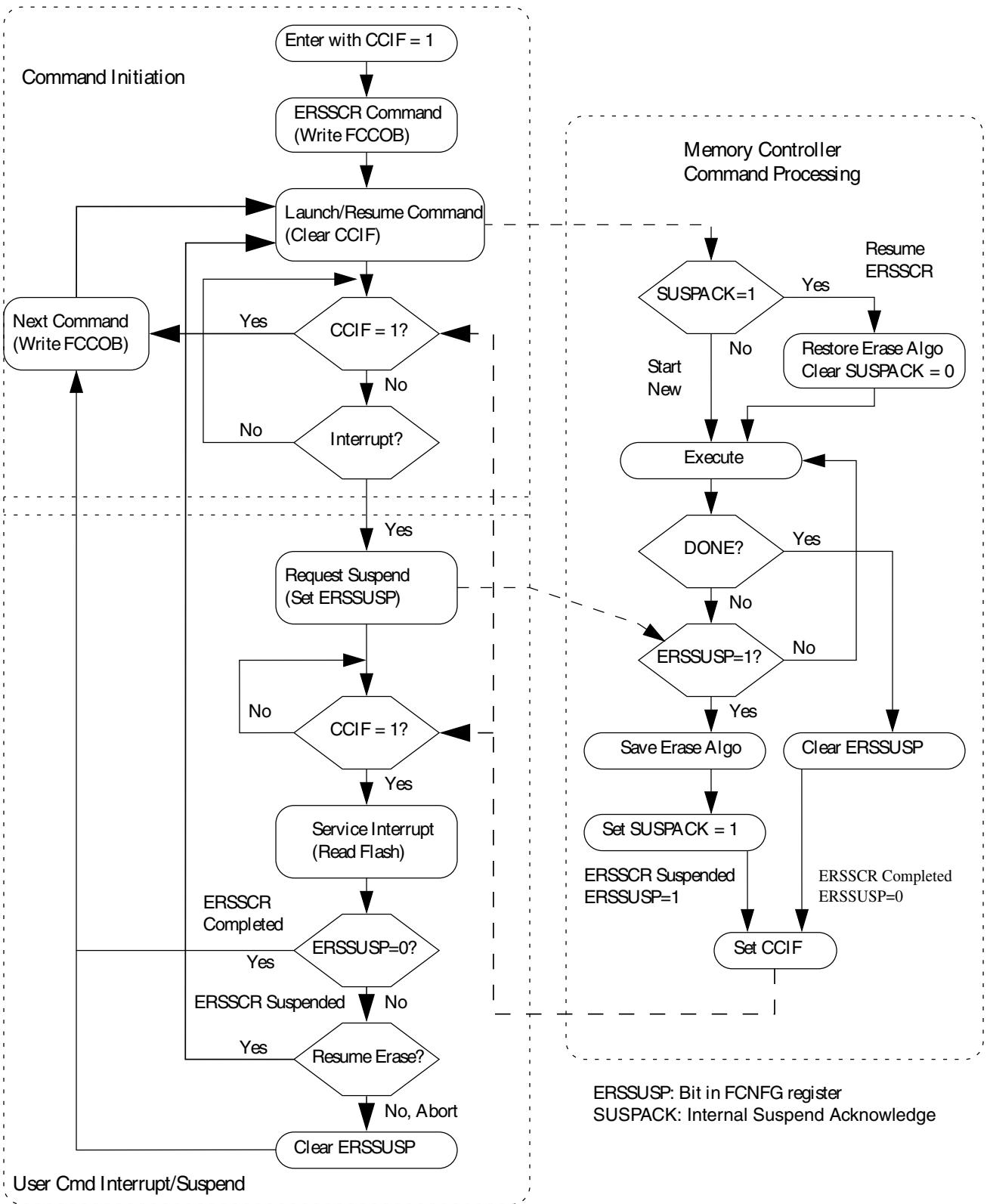


Figure 29-13. Suspend and Resume of Erase Flash Sector Operation

29.4.12.7.4 Impact of radio activity on Erase Flash Sector operation

The RSIM module provides an early indicator to the flash module that the radio is going active and also indicates when the radio goes inactive. If the radio active indicator is asserted while the Erase Flash Sector operation is active, the operation will stall to reduce power consumption before the radio goes active. The Erase Flash Sector operation will resume after the radio 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 radio activity.

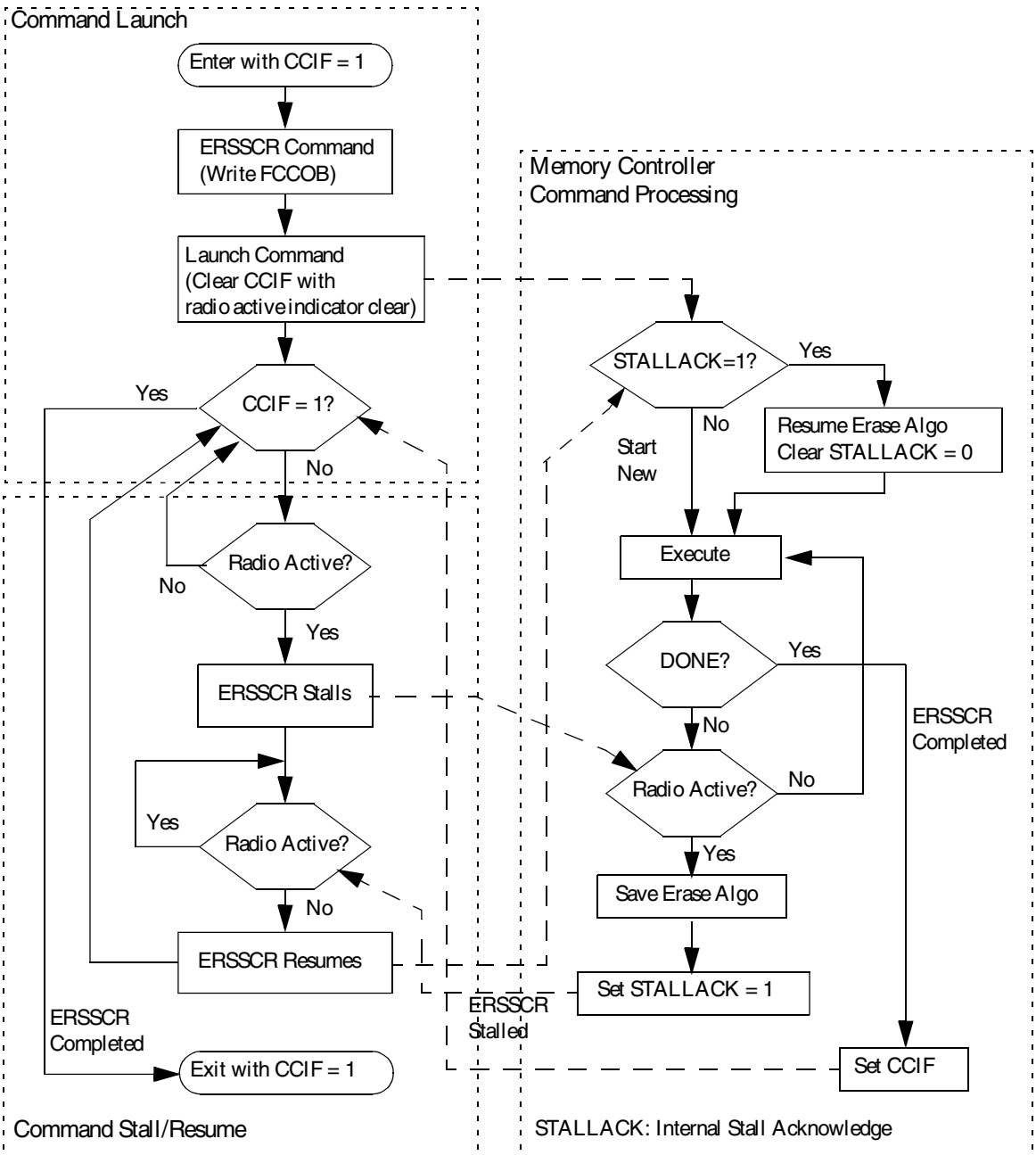


Figure 29-14. Radio Active Impact on Erase Flash Sector Operation

29.4.12.8 Program Section command

The Program Section operation programs the data found in the section program buffer to previously erased locations in the flash memory using an embedded algorithm. Data is preloaded into the section program buffer by writing to the FlexRAM while it is set to function as a programming acceleration RAM (see Flash sector programming).

The section program buffer is limited to the lower quarter of the programming acceleration RAM (relative byte addresses 0x0000-0x07FF - be sure to check your device specific memory map for the location of the programming acceleration RAM or FlexRAM). Data written to the remainder of the programming acceleration RAM is ignored and may be overwritten during Program Section command execution.

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 29-26. Program Section Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x0B (PGMSEC)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Number of phrases to program [15:8]
5	Number of phrases to program [7:0]

1. Must be 64-bit aligned (Flash address [2:0] = 000).

After clearing CCIF to launch the Program Section command, the FTFE will block access to the programming acceleration RAM (program flash only devices) or FlexRAM (FlexNVM devices) and program the data residing in the Section Program Buffer into the flash memory starting at the flash address provided.

The starting address must be unprotected (see the description of the FPROT registers) to permit execution of the Program Section operation. If the swap system is enabled, the phrase containing the swap indicator in each half of the program flash space is implicitly protected from programming. If the phrase containing the swap indicator address is encountered during the Program Section operation, it will be bypassed without setting FPVIOL and the contents will not be programmed. Programming, which is not allowed to cross a flash sector boundary, continues until all requested phrases have been programmed.

After the Program Section operation has completed, the CCIF flag will set and normal access to the FlexRAM is restored. The contents of the Section Program Buffer are not changed by the Program Section operation unless the swap system is enabled and the Program Section operation is targeting the phrase containing the swap indicator in which case that phrase is changed to all ones.

CAUTION

The Program Section operation will not react to the early indicator for radio activity. Therefore, the Program Section command must not be launched if there is a concern about inadequate power available to support both radio activity and the Program Section operation.

Table 29-27. Program Section 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 64-bit aligned	FSTAT[ACCERR]
The requested section crosses a program flash sector boundary	FSTAT[ACCERR]
The requested number of phrases is zero	FSTAT[ACCERR]
The space required to store data for the requested number of phrases is more than one quarter the size of the programming acceleration RAM (program flash only devices) or FlexRAM (FlexNVM devices)	FSTAT[ACCERR]
The FlexRAM is not set to function as a traditional RAM, i.e. set if RAMRDY=0	FSTAT[ACCERR]
The flash address falls in a protected area	FSTAT[FPVIOL]
The requested flash section is located in an XA controlled segment and the Erase All Blocks, Erase All Blocks Unsecure 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]

29.4.12.8.1 Flash sector programming

The process of programming an entire flash sector using the Program Section command is as follows:

1. If required, execute the Set FlexRAM Function command to make the FlexRAM available as traditional RAM and initialize the FlexRAM to all ones.
2. Launch the Erase Flash Sector command to erase the flash sector to be programmed.
3. Beginning with the starting address of the programming acceleration RAM (program flash only devices) or FlexRAM (FlexNVM devices), sequentially write enough data to the RAM to fill an entire flash sector, or as much data is allowed due to RAM size versus flash sector size. This area of the RAM serves as the section program buffer.

NOTE

In step 1, the section program buffer was initialized to all ones, the erased state of the flash memory.

The section program buffer can be written to while the operation launched in step 2 is executing, i.e. while CCIF = 0.

4. Execute the Program Section command to program the contents of the section program buffer into the selected flash sector.
5. Repeat steps 3 through 4 to complete the entire flash sector, if necessary.
6. To program additional flash sectors, repeat steps 2 through 5.
7. To restore EEPROM functionality, execute the Set FlexRAM Function command to make the FlexRAM available for EEPROM.

29.4.12.9 Read 1s All Blocks Command

The Read 1s All Blocks command checks if the program flash blocks, data flash blocks, EEPROM backup records, and data flash IFR 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 29-28. 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 FTFE :

- sets the read margin for 1s according to Table 29-29,
- checks the contents of the program flash, data flash, EEPROM backup records, and data flash IFR are in the erased state.

If the FTFE 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 flash memory resources are not in the fully erased state, the FSTAT[MGSTAT0] bit is set.

The EEERDY and RAMRDY bits are clear during the Read 1s All Blocks operation and are restored at the end of the Read 1s All Blocks operation.

The CCIF flag sets after the Read 1s All Blocks operation has completed.

CAUTION

The Read 1s All Blocks operation will not react to the early indicator for radio activity. Therefore, the Read 1s All Blocks command must not be launched if there is a concern about inadequate power available to support both radio activity and the Read 1s All Blocks operation.

Table 29-29. 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 29-30. Read 1s All Blocks Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid margin choice is specified	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

29.4.12.10 Read Once Command

The Read Once command provides read access to a reserved 96-byte field located in the program flash 0 IFR (see Program flash 0 IFR map and Program Once field). Access to the Program Once field is via 12 records, each 8 bytes long. The Program Once field is programmed using the Program Once command described in Program Once command.

Table 29-31. Read Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x41 (RDONCE)
1	Program Once record index (0x00 - 0x0B)
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
9	Program Once byte 5 value
A	Program Once byte 6 value
B	Program Once byte 7 value

After clearing CCIF to launch the Read Once command, an 8-byte Program Once record is read from the program flash IFR 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 to 0x0B. During execution of the Read Once command, any attempt to read addresses within the program flash block containing this 96-byte field returns invalid data. The Read Once command can be executed any number of times.

Table 29-32. 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]

29.4.12.11 Program Once command

The Program Once command enables programming to a reserved 96-byte field in the program flash 0 IFR (see Program flash 0 IFR map and Program Once field). Access to the Program Once field is via 12 records, each 8 bytes long. The Program Once field can be read using the Read Once command (see Read Once Command) or using the Read Resource command (see Read Resource Command). Each Program Once record can be programmed only once since the program flash 0 IFR cannot be erased.

Table 29-33. Program Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x43 (PGMONCE)
1	Program Once record index (0x00 - 0x0B)
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
9	Program Once Byte 5 value
A	Program Once Byte 6 value
B	Program Once Byte 7 value

After clearing CCIF to launch the Program Once command, the FTFE 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.

The reserved program flash 0 IFR location accessed by the Program Once command cannot be erased and 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 to 0x0B. During execution of the Program Once command, any attempt to read addresses within program flash 0 returns invalid data.

Table 29-34. 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-erased 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_FFFF_FFFF, the Program Once command is allowed to execute again on that same record.

29.4.12.12 Erase All Blocks Command

The Erase All Blocks operation erases all flash memory, initializes the FlexRAM, verifies all memory contents, and releases MCU security.

Table 29-35. 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 FTFE erases all program flash memory, program flash swap IFR space, data flash memory, data flash IFR space, EEPROM backup memory, and FlexRAM, then verifies that all are erased.

If the FTFE verifies that all flash memories and the FlexRAM were properly erased, access control is disabled and security is released by setting the FSEC[SEC] field to the unsecure state and the FCNFG[RAMRDY] bit is set. The Erase All Blocks command aborts if any flash or FlexRAM region is protected. The swap indicator address in the program flash blocks are not implicitly protected from the erase operation. 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 0 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 radio activity. Therefore, the Erase All Blocks command must not be launched if there is a concern about inadequate power available to support both radio activity and the Erase All Blocks operation.

Table 29-36. 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, data flash memory, or FlexRAM 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.

29.4.12.12.1 Triggering an erase all external to the flash module

The functionality of the Erase All Blocks/Erase All Blocks Unsecure 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 FCCOB0 register must not contain 0x44. When invoked, the erase-all function erases all program flash memory, program flash swap IFR space, data flash memory, data flash IFR space, EEPROM backup, and FlexRAM regardless of the state of the FSTAT[ACCERR and FPVIOL] flags or the protection settings or the state of the flash swap system. If the post-erase verify passes, access control determined by the contents of the FXACC registers is disabled and the routine releases security by setting the FSEC[SEC] field register to the unsecure state and the FCNFG[RAMRDY] bit sets. 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, except FPVIOL, is available as described in Erase All Blocks Command/Erase All Blocks Unsecure Command.

CAUTION

Since the IFR Swap Field in the program flash swap IFR containing the swap indicator address is erased during the Erase All Blocks command operation, the swap system becomes uninitialized. The Swap Control command must be run with the initialization code to set the swap indicator address and initialize the swap system.

CAUTION

The Erase All Blocks operation will not react to the early indicator for radio activity. Therefore, the Erase All pin must not be asserted if there is a concern about inadequate power available to support both radio activity and the Erase All Blocks operation.

29.4.12.13 Verify Backdoor Access Key command

Execution of the Verify Backdoor Access Key command is 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. The column labeled Flash Configuration Field offset address shows the location of the matching byte in the Flash Configuration Field.

Table 29-37. 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
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 FTFE checks the FSEC[KEYEN] bits to verify that this command is enabled. If not enabled, the FTFE sets the FSTAT[ACCERR] bit and terminates. If the command is enabled, the FTFE 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 FTFE 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 29-38. Verify Backdoor Access Key Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
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]

29.4.12.14 Swap Control command (program flash only devices)

The Swap Control command handles specific activities associated with swapping the two halves of program flash memory within the memory map.

Table 29-39. Swap Control Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x46 (SWAP)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Swap Control Code: 0x01 - Initialize Swap System 0x02 - Set Swap in Update State 0x04 - Set Swap in Complete State 0x08 - Report Swap Status 0x10 - Disable Swap System
Returned values	
5	Current Swap Mode: 0x00 - Uninitialized 0x01 - Ready 0x02 - Update

Table continues on the next page...

Table 29-39. Swap Control Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents [7:0]
	0x03 - Update-Erased 0x04 - Complete 0x05 - Disabled
6	Current Swap Block Status: 0x00 - Program flash block 0 at 0x0_0000 0x01 - Program flash block 1 at 0x0_0000
7	Next Swap Block Status (after any reset): 0x00 - Program flash block 0 at 0x0_0000 0x01 - Program flash block 1 at 0x0_0000

1. Must be 64-bit aligned (Flash address [2:0] = 000).

Upon clearing CCIF to launch the Swap Control command, the FTFE will handle swap-related activities based on the Swap Control code provided in FCCOB4 as follows:

- 0x01 (Initialize Swap System to UPDATE-ERASED State) - After verifying that the current swap state is UNINITIALIZED, and that both phrases which will contain the swap indicators (located in each half of the Program flash memory within the relative phrase flash address provided) are erased, and that the flash address provided is in the lower half of Program flash memory but not in the Flash Configuration Field, the flash address provided (shifted with bits[2:0] removed) will be programmed into the IFR Swap Field found in the program flash swap IFR. After the swap indicator address has been programmed into the IFR Swap Field, the swap enable word will be programmed to 0x0000. After the swap enable word has been programmed, the swap indicator located in the lower half of the Program flash memory will be programmed to 0xFF00.
- 0x02 (Progress Swap to UPDATE State) - After verifying that the current swap state is READY and that the aligned flash address provided matches the one stored in the IFR Swap Field, the swap indicator located in the currently active program flash block will be programmed to 0xFF00.
- 0x04 (Progress Swap to COMPLETE State) - After verifying that the current swap state is UPDATE-ERASED and that the aligned flash address provided matches the one stored in the IFR Swap Field, the swap indicator located in the currently active program flash block will be programmed to 0x0000. Before executing with this Swap Control code, the user must erase the non-active swap indicator using the Erase Flash Block or Erase Flash Sector commands and update the application code or data as needed.
- 0x08 (Report Swap Status) - After verifying that the aligned flash address provided is in the lower half of Program flash memory but not in the Flash Configuration Field, the status of the swap system will be reported as follows:

- FCCOB5 (Current Swap State) - indicates the current swap state based on the status of the swap disable word, swap enable word and the swap indicators. If the MGSTAT0 flag is set after command completion, the swap state returned was not successfully transitioned from and the appropriate swap command code must be attempted again. If the current swap state is UPDATE and the non-active swap indicator is 0xFFFF, the current swap state is changed to UPDATE-ERASED.
- FCCOB6 (Current Swap Block Status) - indicates which program flash block is currently located at relative flash address 0x0_0000.
- FCCOB7 (Next Swap Block Status) - indicates which program flash block will be located at relative flash address 0x0_0000 after the next reset of the FTFE module.

- 0x10 (Disable Swap System) - After verifying that the current swap state is UNINITIALIZED and that the aligned flash address provided is not in the Flash Configuration Field, the swap disable word, located in the IFR Swap Field, is programmed to 0x0000 and the swap system changed to the DISABLED state with Program flash block 0 located at relative flash address 0x0_0000.

NOTE

It is recommended that the user execute the Swap Control command to report swap status (code 0x08) after any reset to determine if issues with the swap system were detected during the swap state determination procedure.

NOTE

It is recommended that the user write 0xFF to FCCOB5, FCCOB6, and FCCOB7 since the Swap Control command will not always return the swap state and status fields when an ACCERR is detected.

The CCIF flag is set after the Swap Control operation has completed.

The swap indicators are implicitly protected from being programmed during Program Phrase or Program Section command operations and are implicitly unprotected during Swap Control command operations. The swap indicators are implicitly protected from being erased during Erase Flash Block and Erase Flash Sector command operations unless the swap indicator being erased is in the non-active program flash block and the swap system is in the UPDATE or UPDATE-ERASED state. The Erase All Blocks command or erase-all function can be used to place the swap system in the UNINITIALIZED state.

Table 29-40. Swap Control Command Error Handling

Error Condition	Swap Control Code	Error Bit
Command not available in current mode/security[1]	All	FSTAT[ACCERR]
Flash address is not in the lower half of program flash memory	All	FSTAT[ACCERR]
Flash address is in the Flash Configuration Field	1, 8, 10	FSTAT[ACCERR]
Flash address is not 64-bit aligned	All	FSTAT[ACCERR]
Swap system is in the disabled state	1, 2, 4, 10	FSTAT[ACCERR]
Flash address does not match the swap indicator address in the IFR	2, 4	FSTAT[ACCERR]
Swap initialize requested when phrase containing swap indicator (in each half of program flash memory) is not in the erased state	1	FSTAT[ACCERR]
Swap initialize requested when swap system is not in the uninitialized state	1	FSTAT[ACCERR]
Swap update requested when swap system is not in the ready state	2	FSTAT[ACCERR]
Swap complete requested when swap system is not in the update-erased state	4	FSTAT[ACCERR]
Swap disable requested when swap system is not in the uninitialized state	10	FSTAT[ACCERR]
An undefined swap control code is provided	-	FSTAT[ACCERR]
Any errors have been encountered during the swap determination and program-verify operations	1, 2, 4, 10	FSTAT[MGSTAT0]
Any brownouts were detected during the swap determination procedure	8	FSTAT[MGSTAT0]

1. Returned fields will not be updated, i.e. no swap state or status reporting

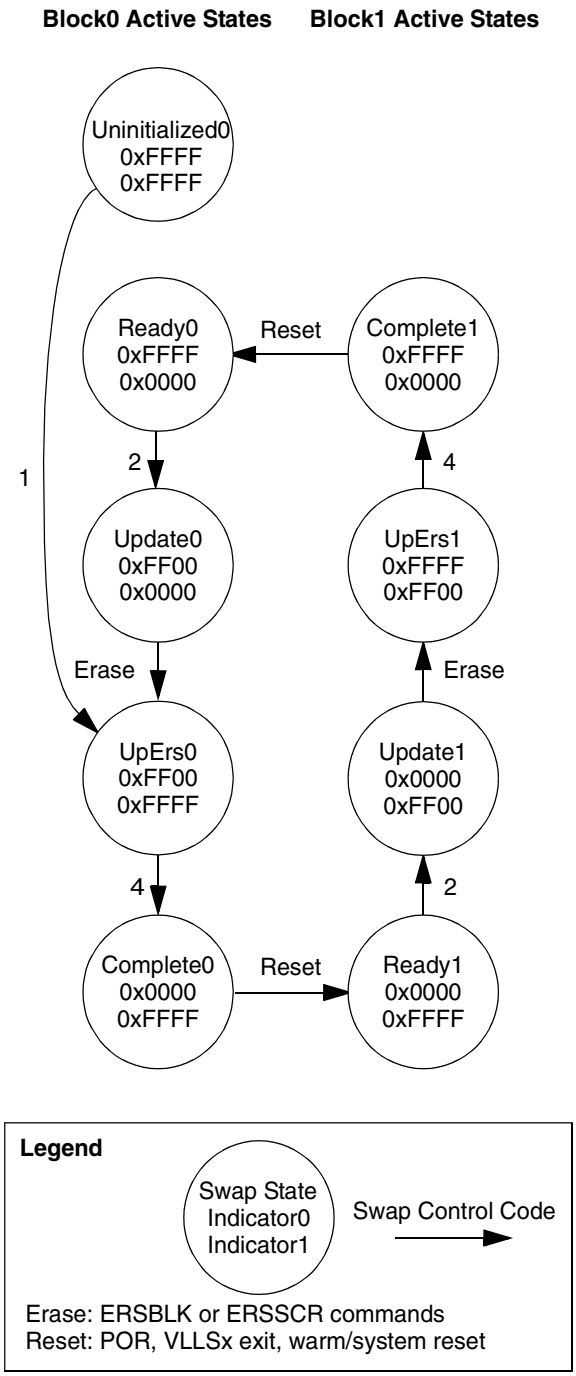


Figure 29-15. Valid Swap State Sequencing

Table 29-41. Swap State Report Mapping

Case	Swap Disable Word	Swap Enable Field[1]	Swap Indicator 0[1]	Swap Indicator 1[1]	Swap State[2]	State Code	MGSTAT 0	Active Block
1	0xFFFF	0xFFFF	-	-	Uninitialized	0	0	0
2	0xFFFF	0x0000	0xFF00	0x0000	Update	2	0	0
3	0xFFFF	0x0000	0xFF00-	0xFFFF	Update-Erased	3	0	0
4	0xFFFF	0x0000	0x0000	0xFFFF[3]	Complete[4]	4	0	0
5	0xFFFF	0x0000	0x0000	0xFFFF	Ready[5]	1	0	1
6	0xFFFF	0x0000	0x0000	0xFF00	Update	2	0	1
7	0xFFFF	0x0000	0xFFFF	0xFF00	Update-Erased	3	0	1
8	0xFFFF	0x0000	0xFFFF[3]	0x0000	Complete[4]	4	0	1
9	0xFFFF	0x0000	0xFFFF	0x0000	Ready[5]	1	0	0
10	0x0000	-	-	-	Disabled	5	0	0
11	0xFFFF	0xXXXX	-	-	Uninitialized	0	1	0
12	0xFFFF	0x0000	0xFFFF	0xFFFF	Uninitialized	0	1	0
13	0xFFFF	0x0000	0xFFXX	0xFFFF	Ready	1	1	0
14	0xFFFF	0x0000	0xFFXX	0x0000	Ready	1	1	0
15[6]	0xFFFF	0x0000	0xXXXX	0x0000	Ready	1	1	0
16[6]	0xFFFF	0x0000	0xFFFF	0xFFXX	Ready	1	1	1
17	0xFFFF	0x0000	0x0000	0xFFXX	Ready	1	1	1
18[6]	0xFFFF	0x0000	0x0000	0xXXXX	Ready	1	1	1
19	0xFFFF	0x0000	0xFF00	0xFFFF[7]	Update	2	1	0
20	0xFFFF	0x0000	0xFF00	0xXXXX	Update	2	1	0
21	0xFFFF	0x0000	0xFF(00)	0xFFXX	Update	2	1	0
22[6]	0xFFFF	0x0000	0x0000	0x0000	Update	2	1	0
23[6]	0xFFFF	0x0000	0xXXXX	0xXXXX	Update	2	1	0
24	0xFFFF	0x0000	0xFFFF[7]	0xFF00	Update	2	1	1
25	0xFFFF	0x0000	0xXXXX	0xFF00	Update	2	1	1
26	0xFFFF	0x0000	0xFFXX	0xFF(00)	Update	2	1	1
27	0xFFFF	0x0000	0xXX00	0xFFFF	Update-Erased	3	1	0
28	0xFFFF	0x0000	0xXXXX	0xFFFF	Update-Erased	3	1	0
29	0xFFFF	0x0000	0xFFFF	0xXX00	Update-Erased	3	1	1
30	0xFFFF	0x0000	0xFFFF	0xXXXX	Update-Erased	3	1	1
31	0xXXXX	-	-	-	Disabled	5	0	0

1. 0xXXXX, 0xFFXX, 0xXX00 indicates a non-valid value was read; 0xFF(00) indicates more 0's than other indicator (if same number of 0's, then swap system defaults to block 0 active)
2. Cases 10-29 due to brownout (abort) detected during program or erase steps related to swap
3. Must read 0xFFFF with erase verify level before transition to Complete allowed
4. No reset since successful Swap Complete execution
5. Reset after successful Swap Complete execution
6. Not a valid case
7. Fails to read 0xFFFF at erase verify level

29.4.12.14.1 Swap state determination

During the reset sequence, the state of the swap system is determined by evaluating the IFR Swap Field in the program flash swap IFR and the swap indicators found within the phrase containing the relative swap indicator address in each half of the program flash memory.

Table 29-42. Program Flash Swap IFR Fields

Address Range	Size (Bytes)	Field Description
0x000 – 0x001	2	Swap Indicator Address
0x002 – 0x003	2	Swap Enable Word
0x004 – 0x009	6	Reserved
0x00A – 0x00B	2	Swap Disable Word
0x00C – 0x3FF	1012	Reserved

29.4.12.15 Erase All Blocks Unsecure Command

The Erase All Blocks Unsecure operation erases all flash memory, initializes the FlexRAM, verifies all memory contents, programs the security byte in the Flash Configuration Field to the unsecure state, and releases MCU security.

Table 29-43. Erase All Blocks Unsecure Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x49 (ERSALLU)

After clearing CCIF to launch the Erase All Blocks Unsecure command, the FTFE erases all program flash memory, program flash swap IFR space, data flash memory, data flash IFR space, EEPROM backup memory, and FlexRAM, then verifies that all are erased.

If the FTFE verifies that all flash memories and the FlexRAM were properly erased, access control is disabled and security is released by setting the FSEC[SEC] field to the unsecure state, the security byte (see Flash configuration field description) is programmed to the unsecure state by the Erase All Blocks Unsecure command, and the FCNFG[RAMRDY] bit is set. The swap indicator address in the program flash blocks are not implicitly protected from the erase operation. If the erase or program verify fails, the FSTAT[MGSTAT0] bit is set. The CCIF flag is set after the Erase All Blocks Unsecure operation completes.

Access control determined by the contents of the FXACC registers will not block execution of the Erase All Blocks Unsecure command. While most Flash memory will be erased, the program flash 0 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 Unsecure command. After completion of the Erase All Blocks Unsecure 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 Unsecure operation will not react to the early indicator for radio activity. Therefore, the Erase All Blocks Unsecure command must not be launched if there is a concern about inadequate power available to support both radio activity and the Erase All Blocks Unsecure operation.

Table 29-44. Erase All Blocks Unsecure Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
Any errors have been encountered during erase or program verify operations	FSTAT[MGSTAT0]

29.4.12.16 Read 1s All Execute-only Segments Command

The Read 1s All Execute-only Segments command checks if the program flash execute-only segments defined by the FXACC registers have been erased to the specified read margin level, if applicable, and releases flash access control if the readout passes, i.e. all data reads as '1'.

Table 29-45. Read 1s All Execute-only Segments Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x4A (RD1XA)
1	Read-1 Margin Choice

After clearing CCIF to launch the Read 1s All Execute-only Segments command, the flash memory module :

- sets the read margin for 1s according to Table 29-46,
- checks the contents of the program flash execute-only segments are in the erased state.

If the flash memory module confirms that these segments are erased, flash access control is disabled until the next reset or, after programming any of the execute-only segments, the Read 1s All Execute-only Segments command is executed and fails with the FSTAT[MGSTAT0] bit set. If the read fails, i.e. all segments are not in the fully erased state, the FSTAT[MGSTAT0] bit is set.

The CCIF flag sets after the Read 1s All Execute-only Segments operation has completed.

CAUTION

The Read 1s All Execute-only Segments operation will not react to the early indicator for radio activity. Therefore, the Read 1s All Execute-only Segments command must not be launched if there is a concern about inadequate power available to support both radio activity and the Read 1s All Execute-only Segments operation.

Table 29-46. Margin Level Choices for Read 1s All Execute-only Segments

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 29-47. Read 1s All Execute-only Segments Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid margin choice is specified	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

29.4.12.17 Erase All Execute-only Segments Command

The Erase All Execute-only Segments operation erases all program flash execute-only segments defined by the FXACC registers, verifies all segments are erased, and releases flash access control.

Table 29-48. Erase All Execute-only Segments Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x4B (ERSXA)

After clearing CCIF to launch the Erase All Execute-only Segments command, the flash memory module erases all program flash execute-only segments, then verifies that all segments are erased.

If the flash memory module verifies that all segments were properly erased, flash access control is disabled until the next reset or, after programming any of the execute-only segments, the Read 1s All Execute-only Segments command is executed and fails with the FSTAT[MGSTAT0] bit set. The Erase All Execute-only Segments command aborts if any XA controlled segment is protected. If the swap system is enabled, the swap indicator address in each program flash block is implicitly protected from erase unless the swap system is in the UPDATE or UPDATE-ERASED state and the program flash segment containing the swap indicator address being erased is in the non-active block. If the erase-verify fails, the FSTAT[MGSTAT0] bit is set. The CCIF flag is set after the Erase All Execute-only Segments operation completes.

Access control determined by the contents of the FXACC registers will not block execution of the Erase All Execute-only Segments command. While all XA controlled segments will be erased, the program flash IFR space containing the Program Once XACC fields will not be erased and, therefore, the contents of the Program Once XACC fields will not change. The contents of the FXACC registers will not be impacted by the execution of the Erase All Execute-only Segments command.

CAUTION

The Erase All Execute-only Segments operation will not react to the early indicator for radio activity. Therefore, the Erase All Execute-only Segments command must not be launched if there is a concern about inadequate power available to support both radio activity and the Erase All Execute-only Segments operation.

Table 29-49. Erase All Execute-only Segments Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
Any XA controlled segment in the program flash memory is protected	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

29.4.12.18 Program Partition command

The Program Partition command prepares the FlexNVM block for use as data flash, EEPROM backup, or a combination of both and initializes the FlexRAM. The Program Partition command must not be launched from flash memory, since flash memory resources are not accessible during Program Partition command execution.

CAUTION

While different partitions of the FlexNVM are available, the intention is that a single partition choice is used throughout the entire lifetime of a given application. The FlexNVM Partition Code choices affect the endurance and data retention characteristics of the device.

Table 29-50. Program Partition Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x80 (PGMPART)
1	Not Used
2	Not Used
3	FlexRAM load during reset option (only bit 0 used): 0 - FlexRAM loaded with valid EEPROM data during reset sequence 1 - FlexRAM not loaded during reset sequence
4	EEPROM Data Set Size Code[1]
5	FlexNVM Partition Code[2]

1. See Valid EEPROM Data Set Size Codes and EEPROM Data Set Size
2. See Valid FlexNVM Partition Codes and FlexNVM partition code

Table 29-51. Valid EEPROM Data Set Size Codes

EEPROM Data Set Size Code (FCCOB4)[1]		EEPROM Data Set Size (Bytes)
EEESPLIT (FCCOB4[5:4)]	EEESIZE (FCCOB4[3:0])	
11	0xF	0[2]
11	0x9	32
11	0x8	64
11	0x7	128
11	0x6	256
11	0x5	512
11	0x4	1,024
11	0x3	2,048
11	0x2	4,096
11	0x1	8,192

1. FCCOB4[7:6] = 00
2. EEPROM Data Set Size must be set to 0 Bytes when the FlexNVM Partition Code is set for no EEPROM.

Table 29-52. Valid FlexNVM Partition Codes

FlexNVM Partition Code DEPART (FCCOB5[3:0])[1]	Data flash Size (Kbytes)	EEPROM-backup Size (Kbytes)
0000	256	0
0011	224	32
0100	192	64
0101	128	128
0110	0	256
1000	0	256
1011	32	224
1100	64	192
1101	128	128
1110	256	0

1. FCCOB5[7:4] = 0000

After clearing CCIF to launch the Program Partition command, the FTFE first verifies that the EEPROM Data Set Size Code and FlexNVM Partition Code in the data flash IFR are erased. If erased, the Program Partition command erases the contents of the FlexNVM memory. If the FlexNVM is to be partitioned for EEPROM backup, the allocated EEPROM backup sectors are formatted for EEPROM use. Finally, the partition codes are programmed into the data flash IFR using the values provided. The Program Partition command also verifies that the partition codes read back correctly after programming. The CCIF flag is set after the Program Partition operation completes.

Prior to launching the Program Partition command, the data flash IFR must be in an erased state, which can be accomplished by executing the Erase All Blocks command or by an external request (see Erase All Blocks Command). The EEPROM Data Set Size Code and FlexNVM Partition Code are read using the Read Resource command (see Read Resource Command).

CAUTION

The Program Partition operation will not react to the early indicator for radio activity. Therefore, the Program Partition command must not be launched if there is a concern about inadequate power available to support both radio activity and the Program Partition operation. If the option is selected to not load the FlexRAM during the reset sequence, the Set FlexRAM Function command must be used with control code 0x00 to load the FlexRAM. However, the Set FlexRAM Function operation will also not react to the early indicator for radio activity. Therefore, if there is a concern about inadequate power

being available, the radio must be inactive for the duration of the Set FlexRAM Function operation that loads the FlexRAM with EEPROM data.

Table 29-53. Program Partition Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
The EEPROM data size and FlexNVM partition code bytes are not initially 0xFFFF	FSTAT[ACCERR]
Invalid EEPROM Data Set Size Code is entered (see Table 29-51 for valid codes)	FSTAT[ACCERR]
Invalid FlexNVM Partition Code is entered (see Table 29-52 for valid codes)	FSTAT[ACCERR]
FlexNVM Partition Code = full data flash (no EEPROM) and EEPROM Data Set Size Code allocates FlexRAM for EEPROM	FSTAT[ACCERR]
FlexNVM Partition Code allocates space for EEPROM backup, but EEPROM Data Set Size Code allocates no FlexRAM for EEPROM	FSTAT[ACCERR]
FCCOB4[7:6] != 00	FSTAT[ACCERR]
FCCOB5[7:4] != 0000	FSTAT[ACCERR]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

29.4.12.19 Set FlexRAM Function command

The Set FlexRAM Function command changes the function of the FlexRAM:

- When not partitioned for EEPROM, the FlexRAM is typically used as traditional RAM.
- When partitioned for EEPROM, the FlexRAM is typically used to store EEPROM data.

Table 29-54. Set FlexRAM Function Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x81 (SETRAM)
1	FlexRAM Function Control Code (see Table 29-55)

Table 29-55. FlexRAM Function Control

FlexRAM Function Control Code	Action
0xFF	Make FlexRAM available as RAM: • Clear the FCNFG[RAMRDY] and FCNFG[EEERDY] flags • Write a background of ones to all FlexRAM locations • Set the FCNFG[RAMRDY] flag
0x00	Make FlexRAM available for EEPROM:

Table 29-55. FlexRAM Function Control

FlexRAM Function Control Code	Action
	• Clear the FCNFG[RAMRDY] and FCNFG[EEERDY] flags • Write a background of ones to all FlexRAM locations • Copy-down existing EEPROM data to FlexRAM • Set the FCNFG[EEERDY] flag

After clearing CCIF to launch the Set FlexRAM Function command, the FTFE sets the function of the FlexRAM based on the FlexRAM Function Control Code.

When making the FlexRAM available as traditional RAM, the FTFE clears the FCNFG[EEERDY] and FCNFG[RAMRDY] flags, overwrites the contents of the entire FlexRAM with a background pattern of all ones, and sets the FCNFG[RAMRDY] flag. The state of the EPROT register does not prevent the FlexRAM from being overwritten. When the FlexRAM is set to function as a RAM, normal read and write accesses to the FlexRAM are available. When large sections of flash memory need to be programmed, e.g. during factory programming, the FlexRAM can be used as the Section Program Buffer for the Program Section command (see Program Section command).

When making the FlexRAM available for EEPROM, the FTFE clears the FCNFG[RAMRDY] and FCNFG[EEERDY] flags, overwrites the contents of the FlexRAM allocated for EEPROM with a background pattern of all ones, and copies the existing EEPROM data from the EEPROM backup record space to the FlexRAM. After completion of the EEPROM copy-down, the FCNFG[EEERDY] flag is set. When the FlexRAM is set to function as EEPROM, normal read and write access to the FlexRAM is available, but writes to the FlexRAM also invoke EEPROM activity.

The CCIF flag will be set after the Set FlexRAM Function operation has completed.

CAUTION

The Set FlexRAM Function operation will not react to the early indicator for radio activity. Therefore, the Set FlexRAM Function command must not be launched with control code 0x00 if there is a concern about inadequate power available to support both radio activity and the Set FlexRAM Function operation.

Table 29-56. Set FlexRAM Function Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
FlexRAM Function Control Code is not defined	FSTAT[ACCERR]
FlexRAM Function Control Code is set to make the FlexRAM available for EEPROM, but FlexNVM is not partitioned for EEPROM	FSTAT[ACCERR]

29.4.13 Security

The FTFE module provides security information to the MCU based on contents of the FSEC security register. The MCU then limits access to FTFE resources as defined in the device's Chip Configuration details. During reset, the FTFE 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. Details of the settings are described in the FSEC register description.

Flash security features are discussed further in AN4507: Using the Kinetis Security and Flash Protection Features . Some features described in the application note may not be available on this device.

Table 29-57. FSEC fields

FSEC field	Description
KEYEN	Backdoor Key Access
MEEN	Mass Erase Capability
FSLACC	Factory Security Level Access
SEC	MCU security

29.4.13.1 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 effect after the next MCU reset.

29.4.13.1.1 Unsecuring the MCU Using Backdoor Key Access

The MCU 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 which 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 MCU. The entire 8-byte key cannot be all 0s or all 1s, i.e. 0x0000_0000_0000_0000 and 0xFFFF_FFFF_FFFF_FFFF 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 MCU can be unsecured by the backdoor key access sequence described below:

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 MCU 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 future use of the Verify Backdoor Access Key command. A reset of the MCU 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 MCU 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 MCU, the security state of the FTFE 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 MCU has full control of the contents of the Flash Configuration Field. The MCU 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.

29.4.13.2 Unsecuring the MCU Using Backdoor Key Access

The MCU 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 which 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 MCU. The entire 8-byte key cannot be all 0s or all 1s, i.e. 0x0000_0000_0000_0000 and 0xFFFF_FFFF_FFFF_FFFF 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 MCU can be unsecured by the backdoor key access sequence described below:

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 MCU 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 future use of the Verify Backdoor Access Key command. A reset of the MCU 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 MCU 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 MCU, the security state of the FTFE 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 MCU has full control of the contents of the Flash Configuration Field. The MCU 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.

29.5 Reset Sequence

On each system reset the FTFE module executes a sequence which establishes initial values for the flash block configuration parameters, FPROT, FDPROT, FEPROT, FOPT, FSEC, FXACC, FSACC, and FACNFG registers and the FCNFG[SWAP, PFLSH, RAMRDY, EEERDY] bits.

CCIF is cleared throughout the reset sequence. The FTFE module holds off all CPU access for a portion of 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 FTFE 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 30
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.

30.1 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.

30.1.1 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

30.1.2 Block diagram

The following figure is the ADC module block diagram.

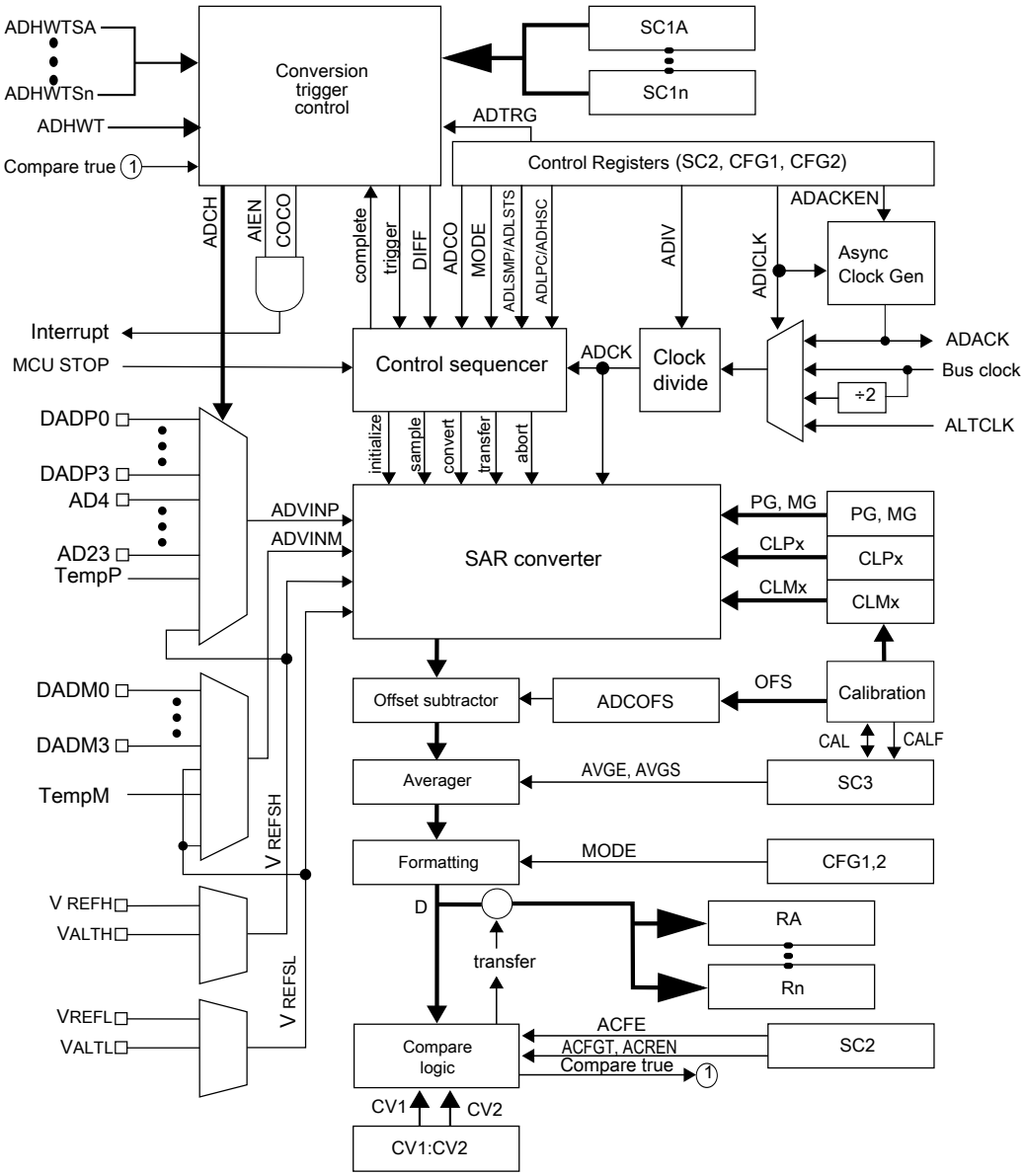


Figure 30-1. ADC block diagram

30.2 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 30-1. 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

30.2.1 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.

30.2.2 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}.

30.2.3 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}.

30.2.4 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.

30.2.5 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.

30.3 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	30.3.1/706
4003_B004	ADC Status and Control Registers 1 (ADC0_SC1B)	32	R/W	0000_001Fh	30.3.1/706
4003_B008	ADC Configuration Register 1 (ADC0_CFG1)	32	R/W	0000_0000h	30.3.2/710
4003_B00C	ADC Configuration Register 2 (ADC0_CFG2)	32	R/W	0000_0000h	30.3.3/711
4003_B010	ADC Data Result Register (ADC0_RA)	32	R	0000_0000h	30.3.4/712

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	30.3.4/712
4003_B018	Compare Value Registers (ADC0_CV1)	32	R/W	0000_0000h	30.3.5/714
4003_B01C	Compare Value Registers (ADC0_CV2)	32	R/W	0000_0000h	30.3.5/714
4003_B020	Status and Control Register 2 (ADC0_SC2)	32	R/W	0000_0000h	30.3.6/715
4003_B024	Status and Control Register 3 (ADC0_SC3)	32	R/W	0000_0000h	30.3.7/717
4003_B028	ADC Offset Correction Register (ADC0_OFS)	32	R/W	0000_0004h	30.3.8/718
4003_B02C	ADC Plus-Side Gain Register (ADC0_PG)	32	R/W	0000_8200h	30.3.9/719
4003_B030	ADC Minus-Side Gain Register (ADC0_MG)	32	R/W	0000_8200h	30.3.10/ 719
4003_B034	ADC Plus-Side General Calibration Value Register (ADC0_CLPD)	32	R/W	0000_000Ah	30.3.11/ 720
4003_B038	ADC Plus-Side General Calibration Value Register (ADC0_CLPS)	32	R/W	0000_0020h	30.3.12/ 721
4003_B03C	ADC Plus-Side General Calibration Value Register (ADC0_CLP4)	32	R/W	0000_0200h	30.3.13/ 721
4003_B040	ADC Plus-Side General Calibration Value Register (ADC0_CLP3)	32	R/W	0000_0100h	30.3.14/ 722
4003_B044	ADC Plus-Side General Calibration Value Register (ADC0_CLP2)	32	R/W	0000_0080h	30.3.15/ 722
4003_B048	ADC Plus-Side General Calibration Value Register (ADC0_CLP1)	32	R/W	0000_0040h	30.3.16/ 723
4003_B04C	ADC Plus-Side General Calibration Value Register (ADC0_CLP0)	32	R/W	0000_0020h	30.3.17/ 723
4003_B054	ADC Minus-Side General Calibration Value Register (ADC0_CLMD)	32	R/W	0000_000Ah	30.3.18/ 724
4003_B058	ADC Minus-Side General Calibration Value Register (ADC0_CLMS)	32	R/W	0000_0020h	30.3.19/ 724
4003_B05C	ADC Minus-Side General Calibration Value Register (ADC0_CLM4)	32	R/W	0000_0200h	30.3.20/ 725
4003_B060	ADC Minus-Side General Calibration Value Register (ADC0_CLM3)	32	R/W	0000_0100h	30.3.21/ 725
4003_B064	ADC Minus-Side General Calibration Value Register (ADC0_CLM2)	32	R/W	0000_0080h	30.3.22/ 726
4003_B068	ADC Minus-Side General Calibration Value Register (ADC0_CLM1)	32	R/W	0000_0040h	30.3.23/ 726
4003_B06C	ADC Minus-Side General Calibration Value Register (ADC0_CLM0)	32	R/W	0000_0020h	30.3.24/ 727

30.3.1 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...

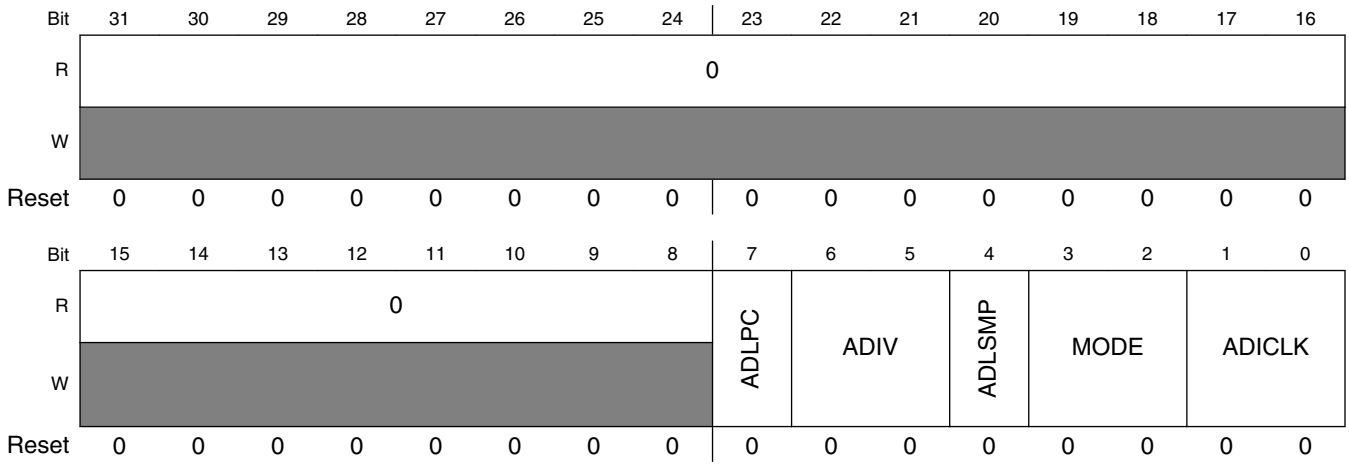
ADC*x*_SC1*n* field descriptions (continued)

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.

30.3.2 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



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)

30.3.3 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.

30.3.4 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 30-2. Data result register description

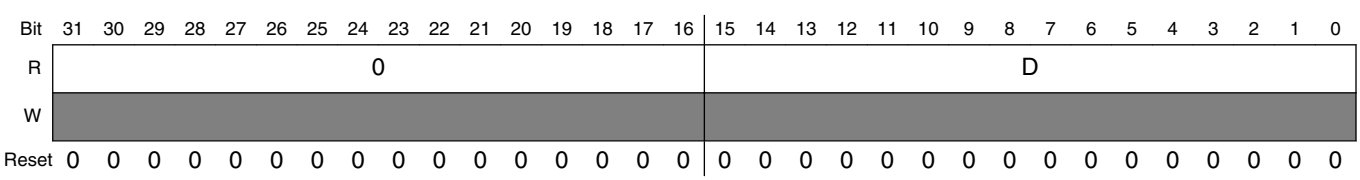
Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
16-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Signed 2's complement
16-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right justified
13-bit differential	S	S	S	S	D	D	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement
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



ADCx_R*n* field descriptions

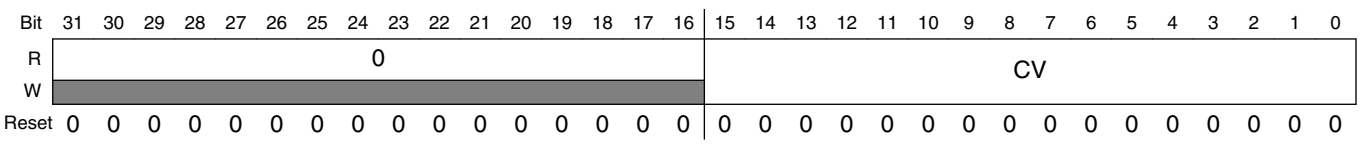
Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
D	Data result

30.3.5 Compare Value Registers (ADCx_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



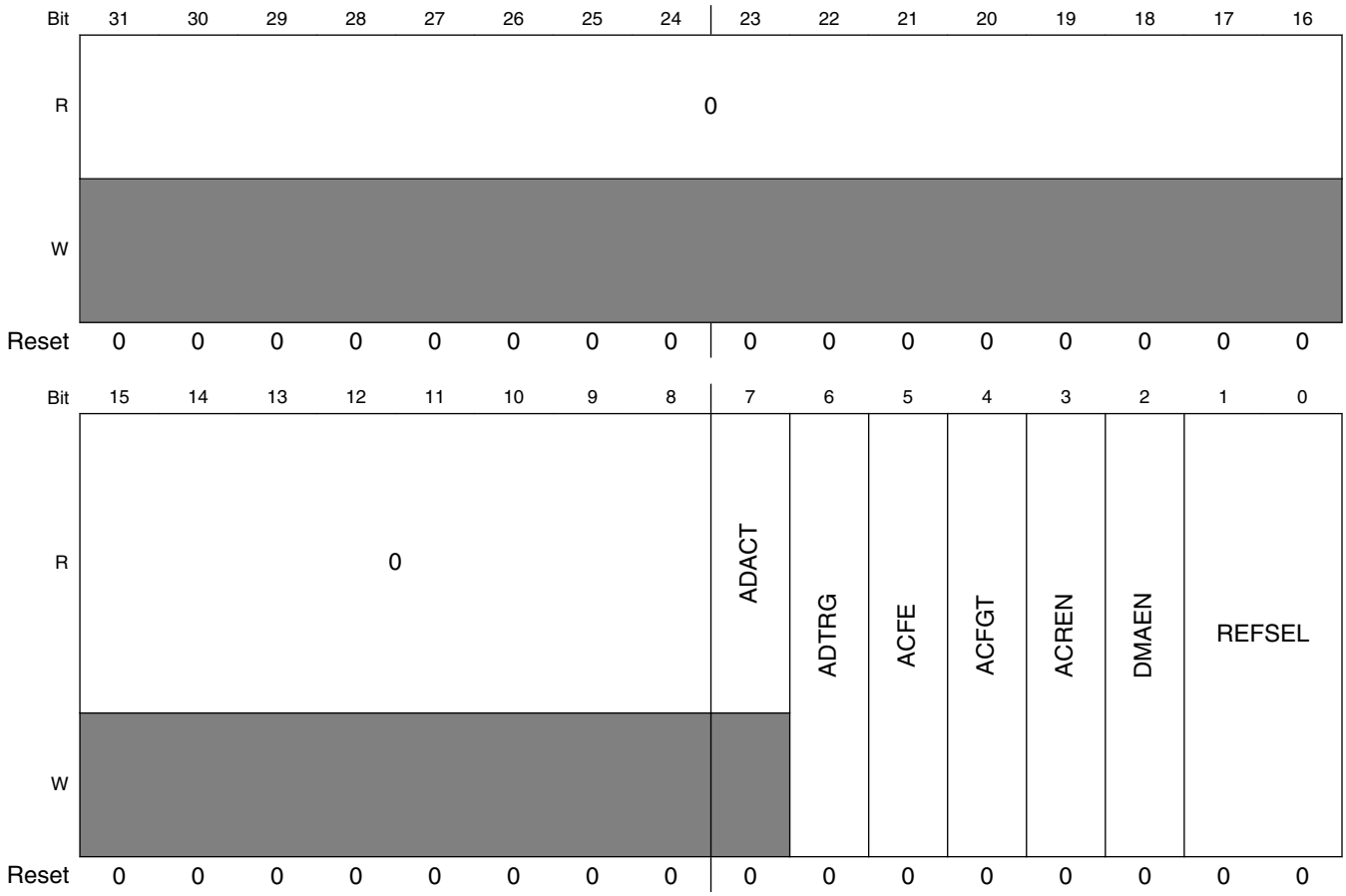
ADCx_CV*n* field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CV	Compare Value.

30.3.6 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



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...

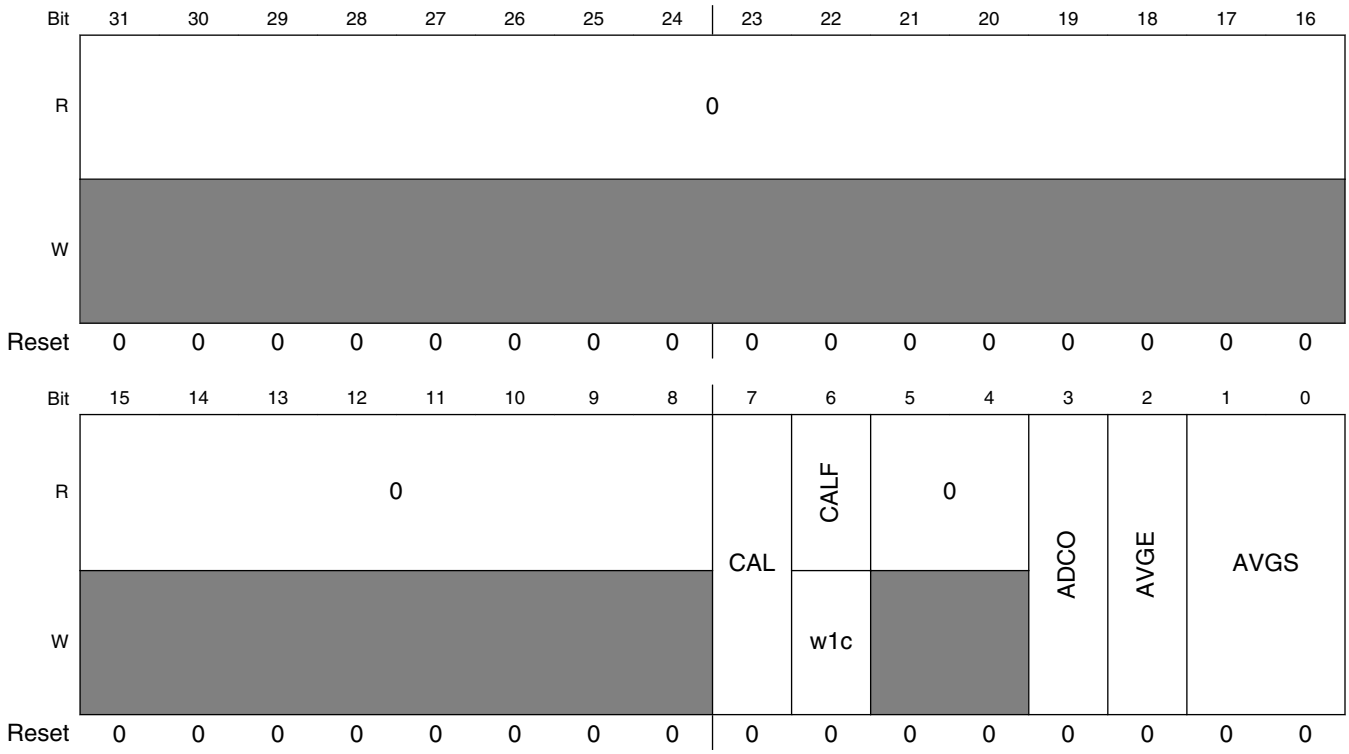
ADCx_SC2 field descriptions (continued)

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

30.3.7 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



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)

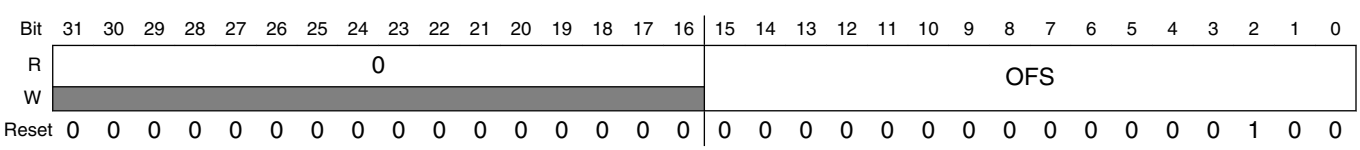
Field	Description
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ADCO	Continuous Conversion Enable Enables continuous conversions. 0 One conversion or one set of conversions if the hardware average function is enabled, that is, AVGE=1, after initiating 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.

30.3.8 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



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

30.3.9 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

30.3.10 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

30.3.11 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

30.3.12 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

30.3.13 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

30.3.14 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

30.3.15 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

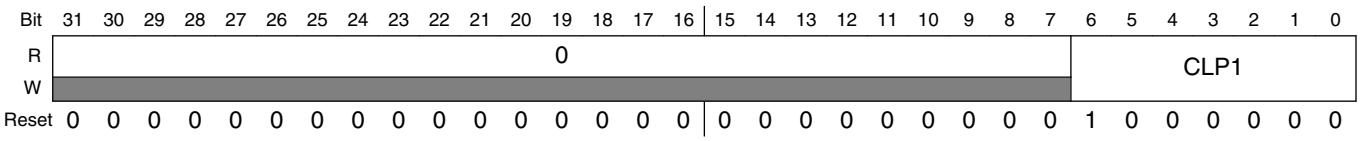
ADCx_CLP2 field descriptions

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

30.3.16 ADC Plus-Side General Calibration Value Register (ADCx_CLP1)

For more information, see CLPD register description.

Address: 4003_B000h base + 48h offset = 4003_B048h



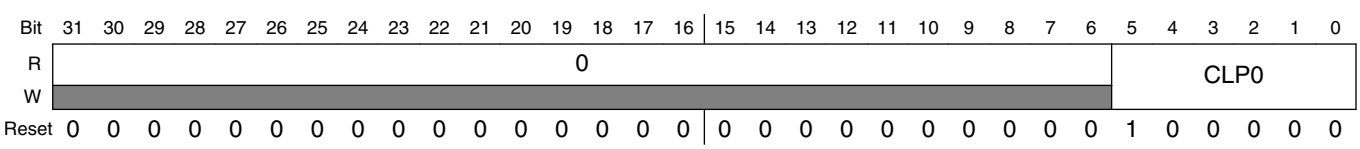
ADCx_CLP1 field descriptions

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

30.3.17 ADC Plus-Side General Calibration Value Register (ADCx_CLP0)

For more information, see CLPD register description.

Address: 4003_B000h base + 4Ch offset = 4003_B04Ch



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

30.3.18 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

30.3.19 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

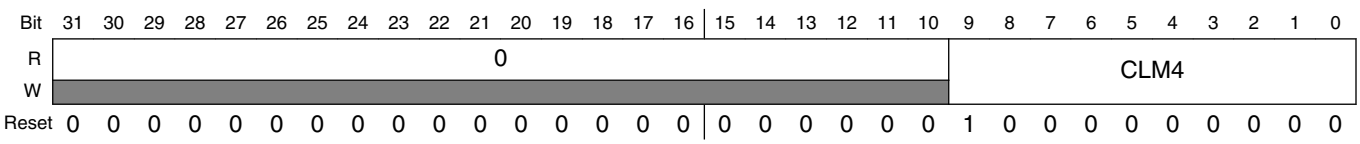
ADCx_CLMS field descriptions

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

30.3.20 ADC Minus-Side General Calibration Value Register (ADCx_CLM4)

For more information, see CLMD register description.

Address: 4003_B000h base + 5Ch offset = 4003_B05Ch



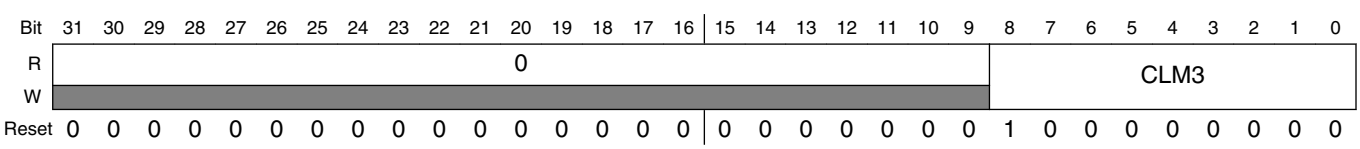
ADCx_CLM4 field descriptions

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

30.3.21 ADC Minus-Side General Calibration Value Register (ADCx_CLM3)

For more information, see CLMD register description.

Address: 4003_B000h base + 60h offset = 4003_B060h



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...

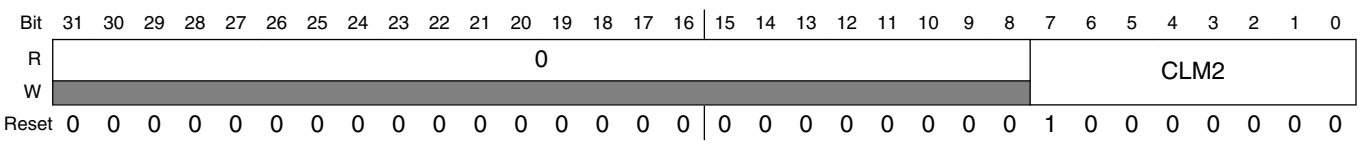
ADCx_CLM3 field descriptions (continued)

Field	Description
CLM3	Calibration Value Calibration Value

30.3.22 ADC Minus-Side General Calibration Value Register (ADCx_CLM2)

For more information, see CLMD register description.

Address: 4003_B000h base + 64h offset = 4003_B064h



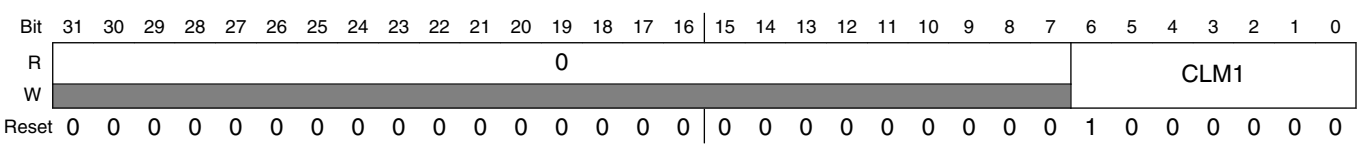
ADCx_CLM2 field descriptions

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

30.3.23 ADC Minus-Side General Calibration Value Register (ADCx_CLM1)

For more information, see CLMD register description.

Address: 4003_B000h base + 68h offset = 4003_B068h



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

30.3.24 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

30.4 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.

30.4.1 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.

30.4.2 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.

30.4.3 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 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.

30.4.4 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

30.4.4.1 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.

30.4.4.2 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.

30.4.4.3 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.

30.4.4.4 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}.

30.4.4.5 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.

$$\text{ConversionTime} = \text{SFCAdder} + \text{AverageNum} \times (\text{BCT} + \text{LSTAdder} + \text{HSCAdder})$$

Equation 1. Conversion time equation

Table 30-3. 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 30-4. 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 30-5. 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 30-5. Base conversion time (BCT) (continued)

Mode	Base conversion time (BCT)
16b single-ended	25 ADCK cycles
16b differential	34 ADCK cycles

Table 30-6. 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 30-7. 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.

30.4.4.6 Conversion time examples

The following examples use the Equation 1 on page 734, and the information provided in Table 30-3 through Table 30-7.

30.4.4.6.1 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 734, and the information provided in Table 30-3 through Table 30-7. The table below lists the variables of Equation 1 on page 734.

Table 30-8. 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.

30.4.4.6.2 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 734, and the information provided in Table 30-3 through Table 30-7. The following table lists the variables of the Equation 1 on page 734.

Table 30-9. 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.

30.4.4.6.3 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 734, and the information provided in Table 30-3 through Table 30-7. The table below lists the variables of Equation 1 on page 734.

Table 30-10. 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.

30.4.4.7 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.

30.4.5 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 30-11. 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.

30.4.6 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.

30.4.7 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.

30.4.8 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.

$$Temp = 25 - \left(\left(V_{TEMP} - V_{TEMP25} \right) \div m \right)$$

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).

30.4.9 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.

30.4.10 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.

30.4.10.1 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.

30.4.10.2 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.

30.4.11 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.

30.5 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 30-6, Table 30-7, and Table 30-8.

Note

Hexadecimal values are designated by a preceding 0x, binary values designated by a preceding %, and decimal values have no preceding character.

30.5.1 ADC module initialization example

30.5.1.1 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.

30.5.1.2 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.
```

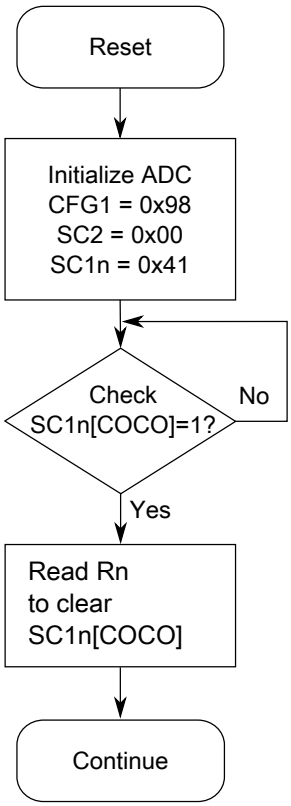


Figure 30-2. Initialization flowchart example

30.6 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.

30.6.1 External pins and routing

30.6.1.1 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.

30.6.1.2 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.

30.6.1.3 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.

30.6.2 Sources of error

30.6.2.1 Sampling error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy.

RAS + RADIN =SC / (FMAX * NUMTAU * CADIN)

Figure 30-3. 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.

30.6.2.2 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 × I_{LEAK} × 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.

30.6.2.3 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.

30.6.2.4 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:

$$1LSB = \left(V_{REFH} \right) / 2^N$$

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.

30.6.2.5 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.

30.6.2.6 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 31
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.

31.1 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.

31.1.1 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

31.1.2 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

31.1.3 ANMUX key features

The ANMUX has the following features:

- Two 8-to-1 channel mux
- Operational over the entire supply range

31.1.4 CMP, DAC and ANMUX diagram

The following figure shows the block diagram for the High-Speed Comparator, DAC, and ANMUX modules.

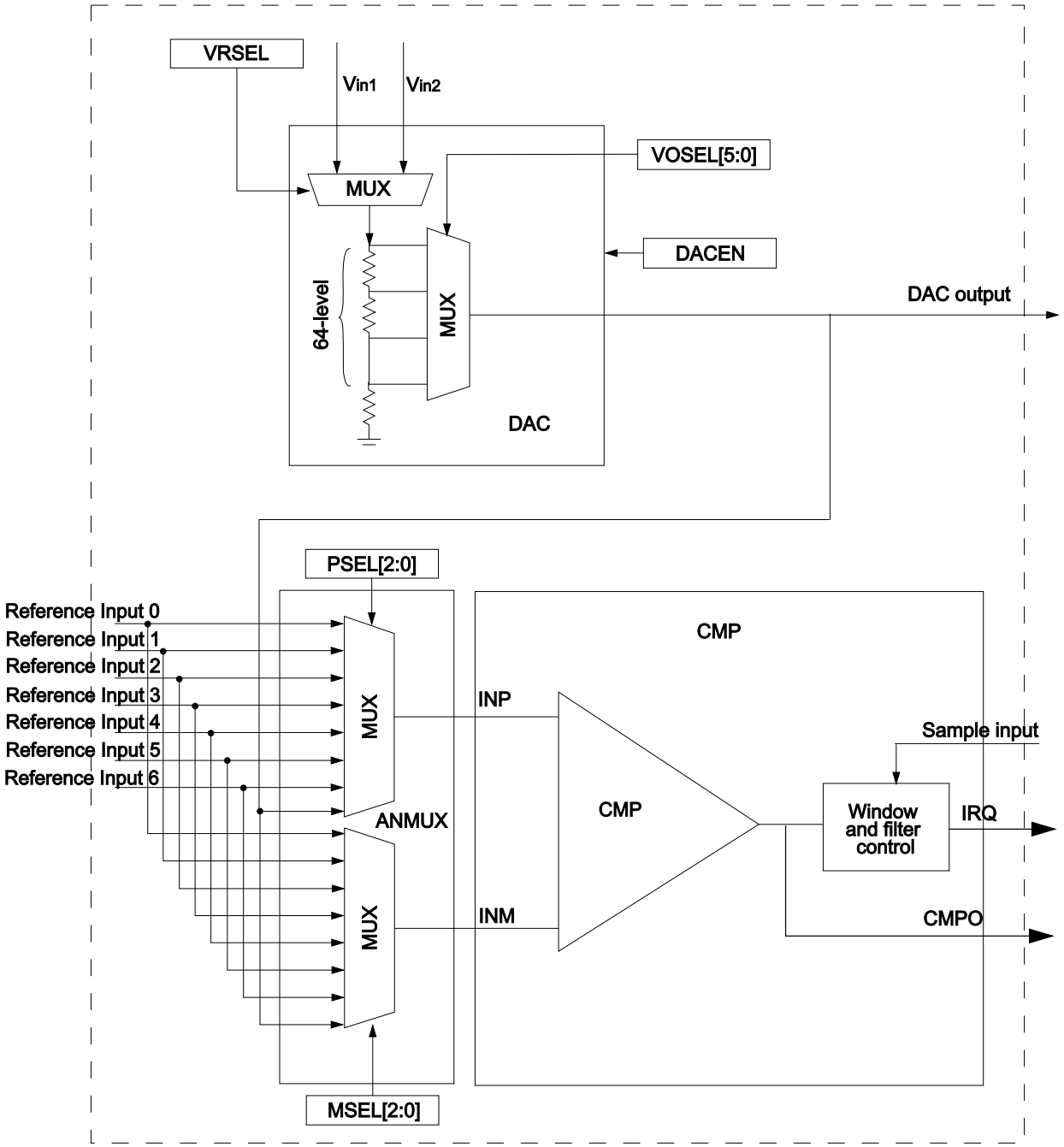


Figure 31-1. CMP, DAC and ANMUX block diagram

31.1.5 CMP block diagram

The following figure shows the block diagram for the CMP module.

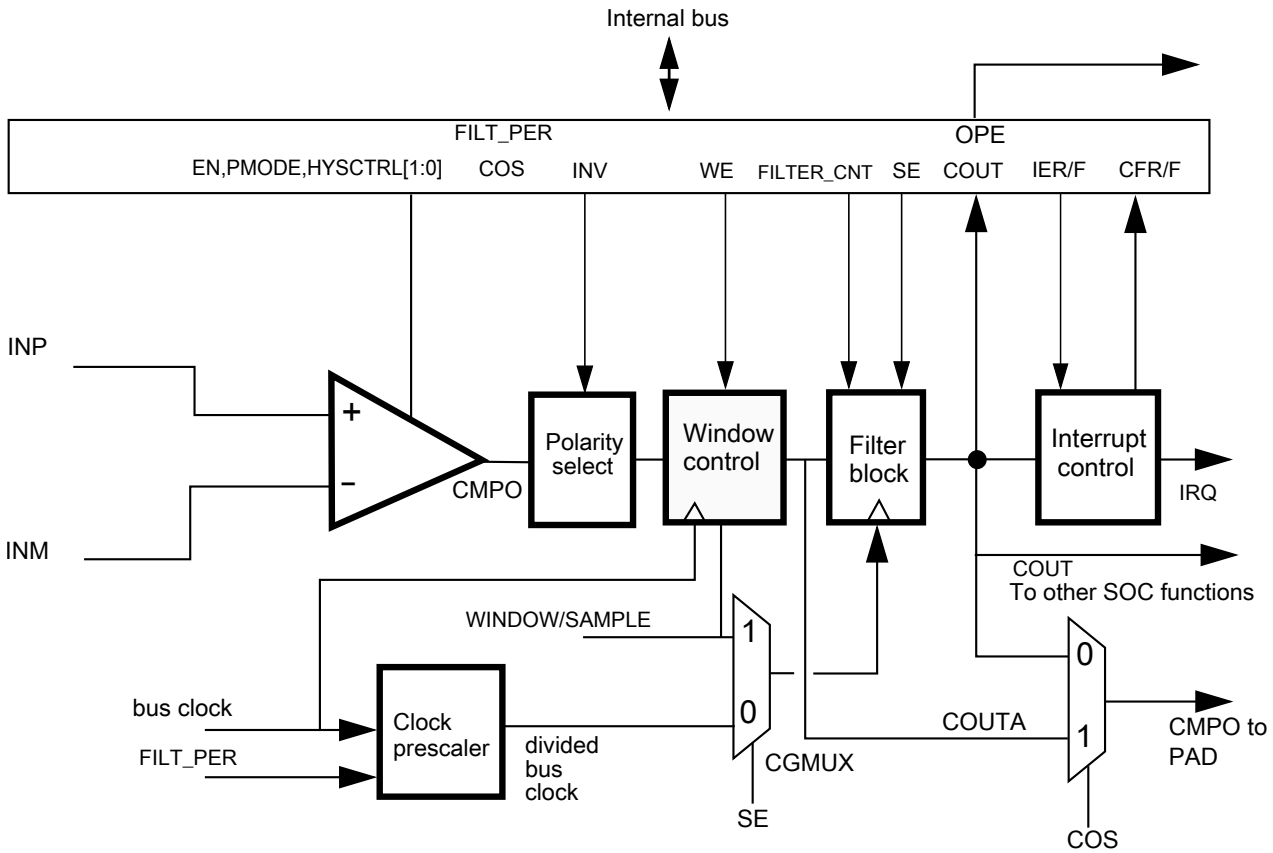


Figure 31-2. Comparator module block diagram

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.

31.2 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	31.2.1/760
4007_3001	CMP Control Register 1 (CMP0_CR1)	8	R/W	00h	31.2.2/761
4007_3002	CMP Filter Period Register (CMP0_FPR)	8	R/W	00h	31.2.3/762
4007_3003	CMP Status and Control Register (CMP0_SCR)	8	R/W	00h	31.2.4/763
4007_3004	DAC Control Register (CMP0_DACCR)	8	R/W	00h	31.2.5/764
4007_3005	MUX Control Register (CMP0_MUXCR)	8	R/W	00h	31.2.6/764

31.2.1 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

31.2.2 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.

31.2.3 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.

31.2.4 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.

31.2.5 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} .

31.2.6 MUX Control Register (CMPx_MUXCR)

Address: 4007_3000h base + 5h offset = 4007_3005h

Bit	7	6	5	4	3	2	1	0
Read Write	PSTM	0	PSEL			MSEL		
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

31.3 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].

31.3.1 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 31-1. 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.

31.3.1.1 Disabled mode (# 1)

In Disabled mode, the analog comparator is non-functional and consumes no power. CMPO is 0 in this mode.

31.3.1.2 Continuous mode (#s 2A & 2B)

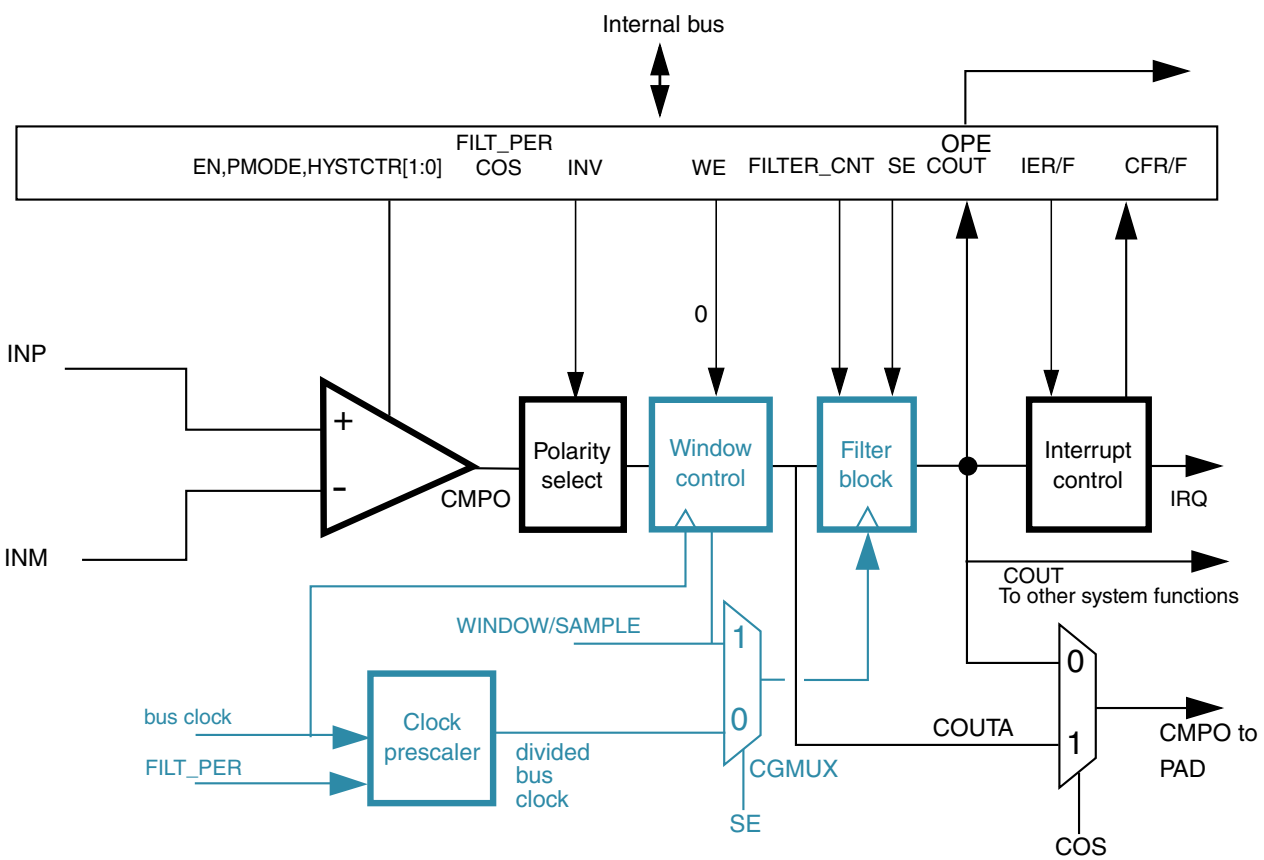


Figure 31-3. Comparator operation in Continuous mode

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.

31.3.1.3 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).

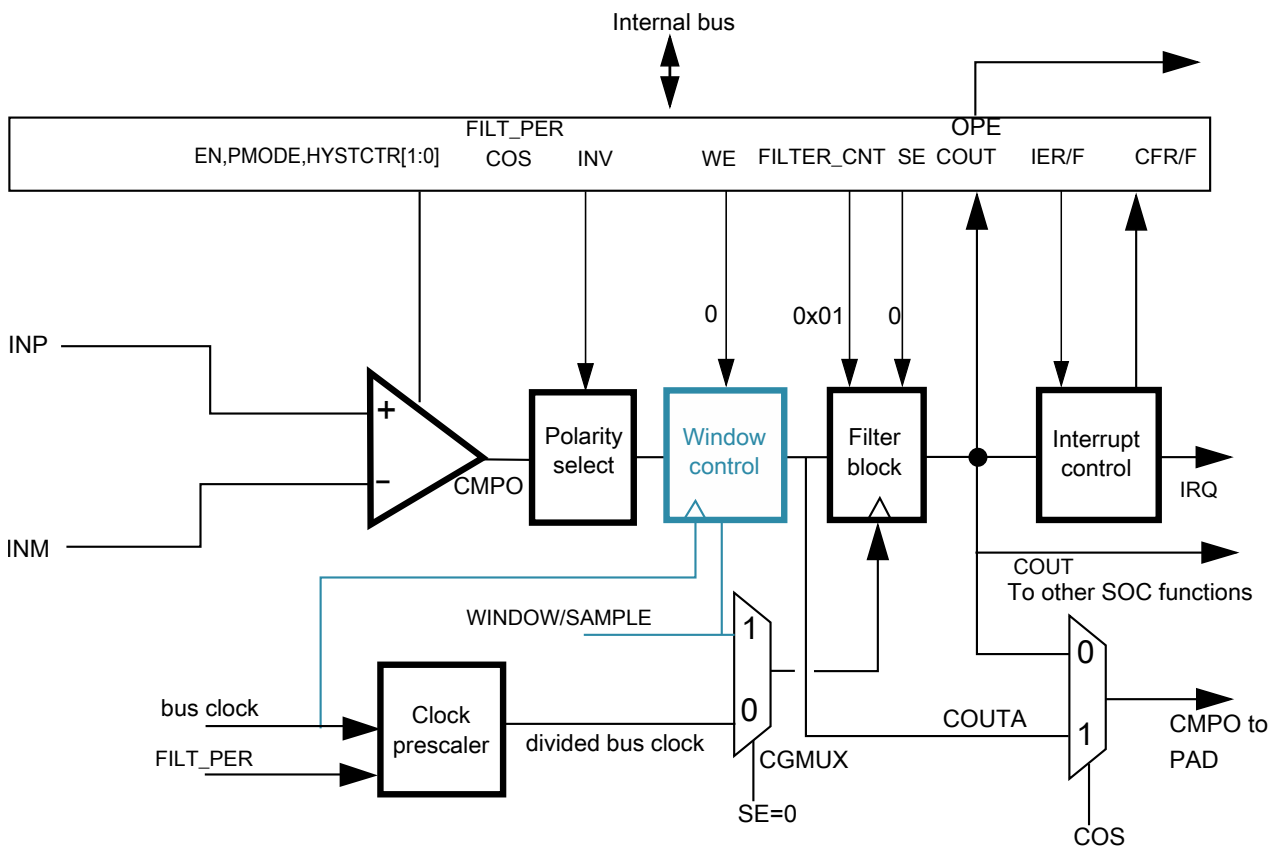


Figure 31-4. Sampled, Non-Filtered (# 3B): sampling interval internally derived

31.3.1.4 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.

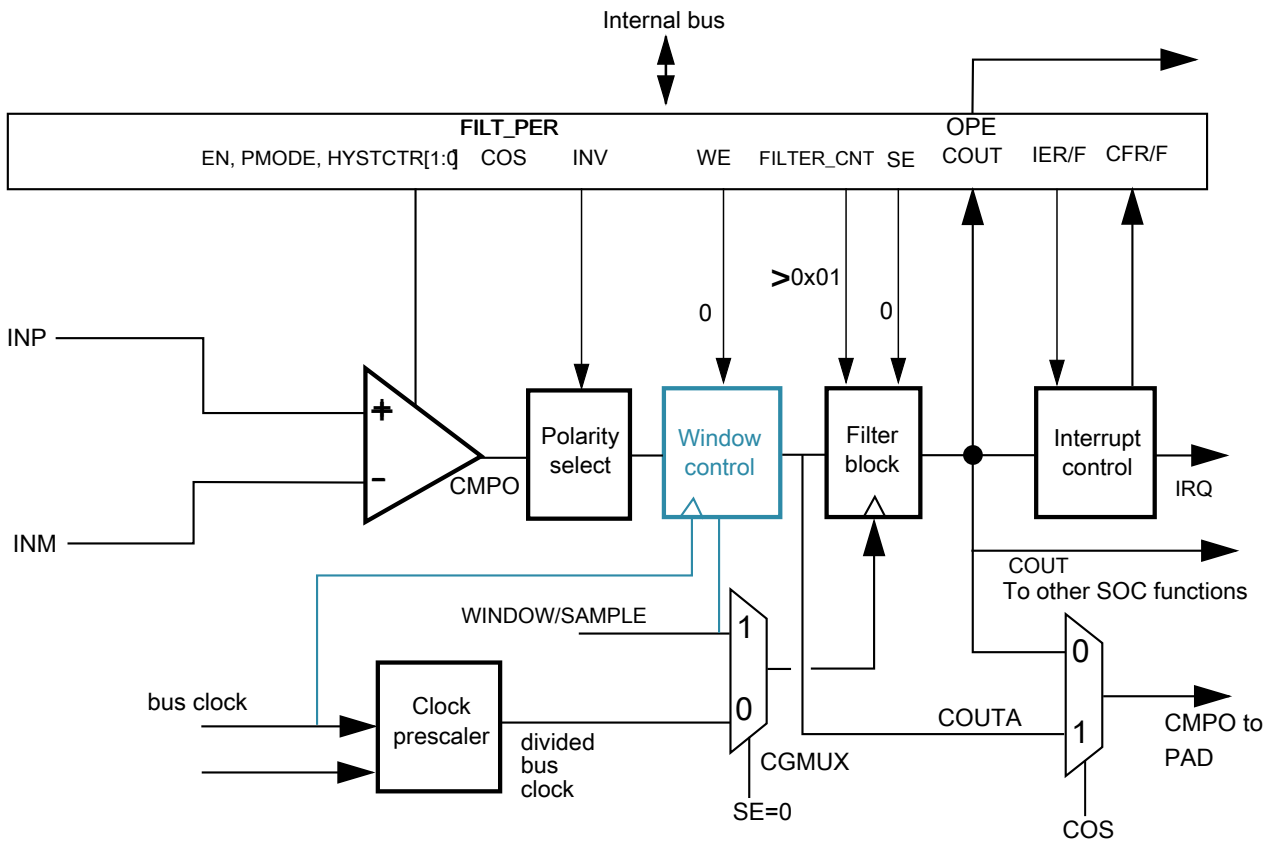


Figure 31-5. Sampled, Filtered (# 4B): sampling point internally derived

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.

31.3.2 Power modes

31.3.2.1 Wait mode operation

During Wait and VLPW modes, the CMP, if enabled, continues to operate normally and a CMP interrupt can wake the MCU.

31.3.2.2 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.

31.3.2.3 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].

31.3.2.4 Background Debug Mode Operation

When the microcontroller is in active background debug mode, the CMP continues to operate normally.

31.3.3 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.

31.3.4 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.

31.3.4.1 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.

31.3.4.2 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 31-2. 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.

31.4 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

31.5 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.

31.6 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.

31.7 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.

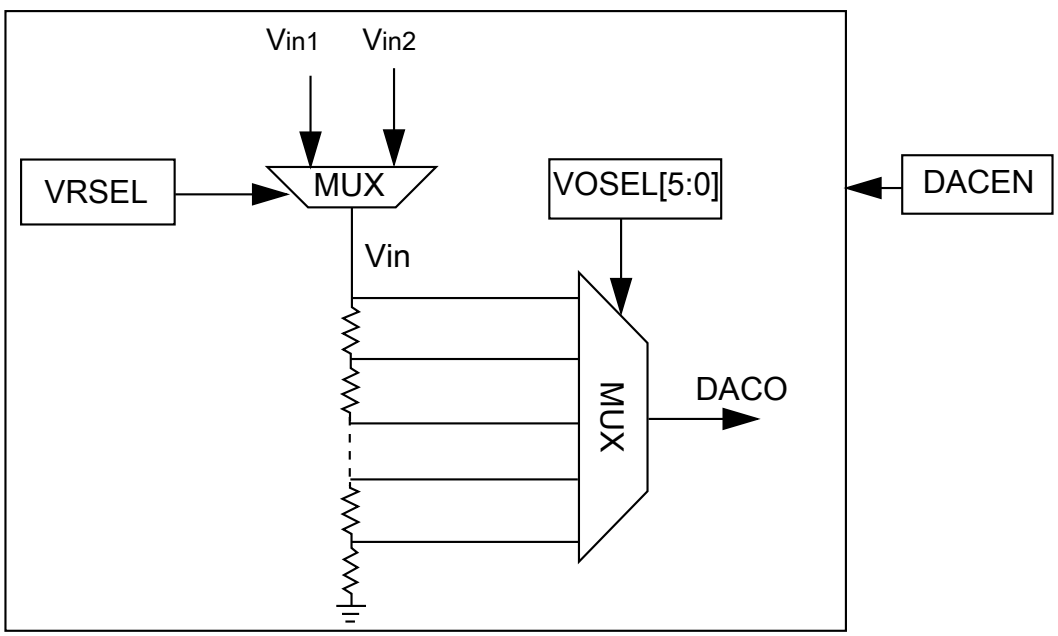


Figure 31-6. 6-bit DAC block diagram

31.8 DAC functional description

This section provides DAC functional description information.

31.8.1 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

31.9 DAC resets

This module has a single reset input, corresponding to the chip-wide peripheral reset.

31.10 DAC clocks

This module has a single clock input, the bus clock.

31.11 DAC interrupts

This module has no interrupts.

31.12 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 32
Voltage Reference(VREF)

32.1 Introduction

The Voltage Reference (VREF) is intended to supply an accurate voltage output that can be trimmed in 0.5 mV steps. The VREF can be used in applications to provide a reference voltage to external devices or used internally as a reference to analog peripherals such as the ADC, DAC, or CMP. The voltage reference has three operating modes that provide different levels of supply rejection and power consumption.

The following figure is a block diagram of the Voltage Reference.

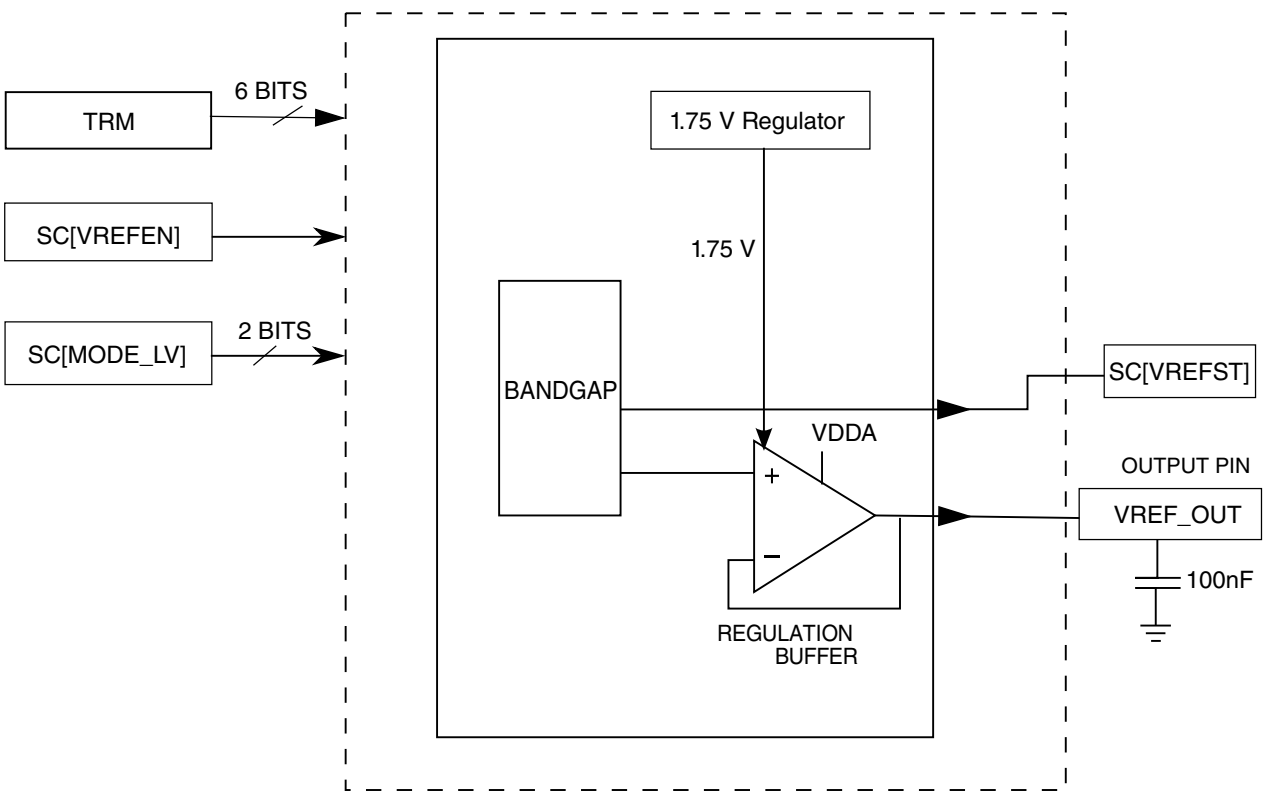


Figure 32-1. Voltage reference block diagram

32.1.1 Overview

The Voltage Reference provides a buffered reference voltage for use as an external reference. In addition, the buffered reference is available internally for use with on chip peripherals such as ADCs and DACs. Refer to the chip configuration details for a description of these options. The reference voltage signal is output when the VREF is enabled. The Voltage Reference output can be trimmed with a resolution of 0.5mV by means of the TRM register TRIM[5:0] bitfield.

32.1.2 Features

The Voltage Reference has the following features:

- Programmable trim register with 0.5 mV steps, automatically loaded with factory trimmed value upon reset
- Programmable buffer mode selection:
  - Off
  - Bandgap enabled/standby (output buffer disabled)
  - Low power buffer mode (output buffer enabled)
  - High power buffer mode (output buffer enabled)
- 1.195 V output at room temperature

32.1.3 Modes of Operation

The Voltage Reference continues normal operation in Run, Wait, and Stop modes. The Voltage Reference can also run in Very Low Power Run (VLPR), Very Low Power Wait (VLPW) and Very Low Power Stop (VLPS). If it is desired to use the VREF regulator and/or the chop oscillator in the very low power modes, the system reference voltage (also referred to as the bandgap voltage reference) must be enabled in these modes. Refer to the chip configuration details for information on enabling this mode of operation. Having the VREF regulator enabled does increase current consumption. In very low power modes it may be desirable to disable the VREF regulator to minimize current consumption. Note however that the accuracy of the output voltage will be reduced (by as much as several mVs) when the VREF regulator is not used.

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.

32.1.4 VREF Signal Descriptions

The following table shows the Voltage Reference signals properties.

Table 32-1. VREF Signal Descriptions

Signal	Description	I/O
VREF_OUT	Internally-generated Voltage Reference output	O

NOTE

When the VREF output buffer is disabled, the status of the VREF_OUT signal is high-impedence.

32.2 Memory Map and Register Definition

VREF memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_4000	VREF Trim Register (VREF_TRM)	8	R/W	See section	32.2.1/779
4007_4001	VREF Status and Control Register (VREF_SC)	8	R/W	00h	32.2.2/780

32.2.1 VREF Trim Register (VREF_TRM)

This register contains bits that contain the trim data for the Voltage Reference.

Address: 4007_4000h base + 0h offset = 4007_4000h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	CHOPEN	TRIM					
Reset	x*	0	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

VREF_TRM field descriptions

Field	Description
7 Reserved	This field is reserved. Upon reset this value is loaded with a factory trim value.
6 CHOPEN	Chop oscillator enable. When set, internal chopping operation is enabled and the internal analog offset will be minimized. This bit is set during factory trimming of the VREF voltage. This bit should be written to 1 to achieve the performance stated in the data sheet. If the internal voltage regulator is being used (REGEN bit is set to 1), the chop oscillator must also be enabled. If the chop oscillator is to be used in very low power modes, the system (bandgap) voltage reference must also be enabled. See the chip-specific VREF information (also known as "chip configuration" details) for a description of how this can be achieved. 0 Chop oscillator is disabled. 1 Chop oscillator is enabled.
TRIM	Trim bits These bits change the resulting VREF by approximately ± 0.5 mV for each step. **NOTE:** Min = minimum and max = maximum voltage reference output. For minimum and maximum voltage reference output values, refer to the Data Sheet for this chip. 000000 Min .... 111111 Max

32.2.2 VREF Status and Control Register (VREF_SC)

This register contains the control bits used to enable the internal voltage reference and to select the buffer mode to be used.

Address: 4007_4000h base + 1h offset = 4007_4001h

Bit	7	6	5	4	3	2	1	0
Read	VREFEN	REGEN	ICOMPEN	0	0	VREFST	MODE_LV	
Write								
Reset	0	0	0	0	0	0	0	0

VREF_SC field descriptions

Field	Description
7 VREFEN	Internal Voltage Reference enable This bit is used to enable the bandgap reference within the Voltage Reference module. **NOTE:** After the VREF is enabled, turning off the clock to the VREF module via the corresponding clock gate register will not disable the VREF. VREF must be disabled via this VREFEN bit.

Table continues on the next page...

VREF_SC field descriptions (continued)

Field	Description
	0 The module is disabled. 1 The module is enabled.
6 REGEN	Regulator enable This bit is used to enable the internal 1.75 V regulator to produce a constant internal voltage supply in order to reduce the sensitivity to external supply noise and variation. If it is desired to keep the regulator enabled in very low power modes, refer to the Chip Configuration details for a description on how this can be achieved. This bit should be written to 1 to achieve the performance stated in the data sheet. **NOTE:** See section Internal voltage regulator for details on the required sequence to enable the internal regulator. 0 Internal 1.75 V regulator is disabled. 1 Internal 1.75 V regulator is enabled.
5 ICOMPEN	Second order curvature compensation enable This bit should be written to 1 to achieve the performance stated in the data sheet. 0 Disabled 1 Enabled
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 VREFST	Internal Voltage Reference stable This bit indicates that the bandgap reference within the Voltage Reference module has completed its startup and stabilization. **NOTE:** This bit is valid only when the chop oscillator is not being used. 0 The module is disabled or not stable. 1 The module is stable.
MODE_LV	Buffer Mode selection These bits select the buffer modes for the Voltage Reference module. 00 Bandgap on only, for stabilization and startup 01 High power buffer mode enabled 10 Low-power buffer mode enabled 11 Reserved

32.3 Functional Description

The Voltage Reference is a bandgap buffer system. Unity gain amplifiers are used.

The VREF_OUT signal can be used by both internal and external peripherals in low and high power buffer mode. A 100 nF capacitor must always be connected between VREF_OUT and VSSA if the VREF is being used.

The following table shows all possible function configurations of the Voltage Reference.

Table 32-2. Voltage Reference function configurations

SC[VREFEN]	SC[MODE_LV]	Configuration	Functionality
0	X	Voltage Reference disabled	Off
1	00	Voltage Reference enabled, bandgap on only	Startup and standby
1	01	Voltage Reference enabled, high-power buffer on	VREF_OUT available for internal and external use. 100 nF capacitor is required.
1	10	Voltage Reference enabled, low power buffer on	VREF_OUT available for internal and external use. 100 nF capacitor is required.
1	11	Reserved	Reserved

32.3.1 Voltage Reference Disabled, SC[VREFEN] = 0

When SC[VREFEN] = 0, the Voltage Reference is disabled, the VREF bandgap and the output buffers are disabled. The Voltage Reference is in off mode.

32.3.2 Voltage Reference Enabled, SC[VREFEN] = 1

When SC[VREFEN] = 1, the Voltage Reference is enabled, and different modes should be set by the SC[MODE_LV] bits.

32.3.2.1 SC[MODE_LV]=00

The internal VREF bandgap is enabled to generate an accurate 1.2 V output that can be trimmed with the TRM register's TRIM[5:0] bitfield. The bandgap requires some time for startup and stabilization. SC[VREFST] can be monitored to determine if the stabilization and startup is complete when the chop oscillator is not enabled.

If the chop oscillator is being used, the internal bandgap reference voltage settles within the chop oscillator start up time, Tchop_osc_stup.

The output buffer is disabled in this mode, and there is no buffered voltage output. The Voltage Reference is in standby mode. If this mode is first selected and the low power or high power buffer mode is subsequently enabled, there will be a delay before the buffer output is settled at the final value. This is the buffer start up delay (Tstup) and the value is specified in the appropriate device data sheet.

32.3.2.2 SC[MODE_LV] = 01

The internal VREF bandgap is on. The high power buffer is enabled to generate a buffered 1.2 V voltage to VREF_OUT. It can also be used as a reference to internal analog peripherals such as an ADC channel or analog comparator input.

If this mode is entered from the standby mode (SC[MODE_LV] = 00, SC[VREFEN] = 1) there will be a delay before the buffer output is settled at the final value. This is the buffer start up delay (Tstup) and the value is specified in the appropriate device data sheet. If this mode is entered when the VREF module is enabled then you must wait the longer of Tstup or until SC[VREFST] = 1 when the chop oscillator is not enabled. If the chop oscillator is being used, you must wait the time specified by Tchop_osc_stup (chop oscillator start up time) to ensure the VREF output has stabilized.

In this mode, a 100 nF capacitor is required to connect between the VREF_OUT pin and VSSA.

32.3.2.3 SC[MODE_LV] = 10

The internal VREF bandgap is on. The low power buffer is enabled to generate a buffered 1.2 V voltage to VREF_OUT. It can also be used as a reference to internal analog peripherals such as an ADC channel or analog comparator input.

If this mode is entered from the standby mode (SC[MODE_LV] = 00, SC[VREFEN] = 1) there will be a delay before the buffer output is settled at the final value. This is the buffer start up delay (Tstup) and the value is specified in the appropriate device data sheet. If this mode is entered when the VREF module is enabled then you must wait the longer of Tstup or until SC[VREFST] = 1 when the chop oscillator is not enabled. If the chop oscillator is being used, you must wait the time specified by Tchop_osc_stup (chop oscillator start up time) to ensure the VREF output has stabilized.

In this mode, a 100 nF capacitor is required to connect between the VREF_OUT pin and VSSA.

32.3.2.4 SC[MODE_LV] = 11

Reserved

32.4 Internal voltage regulator

The VREF module contains an internal voltage regulator that can be enabled to provide additional supply noise rejection. It is recommended that when possible, this regulator be enabled to provide the optimum VREF performance.

If the internal voltage regulator is being used, the chop oscillator must also be enabled. A specific sequence must be followed when enabling the internal regulator as follows:

1. Enable the chop oscillator (VREF_TRM[CHOPEN] = 1)
2. Configure the VREF_SC register to the desired settings with the internal regulator disabled, VREF_SC[REGEN] = 0
3. Wait > 300ns
4. Enable the internal regulator by setting VREF_SC[REGEN] to 1

32.5 Initialization/Application Information

The Voltage Reference requires some time for startup and stabilization. After SC[VREFEN] = 1, SC[VREFST] can be monitored to determine if the stabilization and startup is completed when the chop oscillator is not enabled. When the chop oscillator is enabled, the settling time of the internal bandgap reference is defined by Tchop_osc_stup (chop oscillator start up time). You must wait this time (Tchop_osc_stup) after the internal bandgap has been enabled to ensure the VREF internal reference voltage has stabilized.

When the Voltage Reference is already enabled and stabilized, changing SC[MODE_LV] will not clear SC[VREFST] but there will be some startup time before the output voltage at the VREF_OUT pin has settled. This is the buffer start up delay (Tstup) and the value is specified in the appropriate device data sheet. Also, there will be some settling time when a step change of the load current is applied to the VREF_OUT pin. When the 1.75V VREF regulator is disabled, the VREF_OUT voltage will be more sensitive to supply voltage variation. It is recommended to use this regulator to achieve optimum VREF_OUT performance.

The TRM[CHOPEN], SC[REGEN] and SC[ICOMPEN] bits must be written to 1 to achieve the performance stated in the device data sheet.

NOTE

See section "Internal voltage regulator" for details on the required sequence to enable the internal regulator.

Chapter 33
Timer/PWM Module (TPM)

33.1 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 .

33.1.1 TPM Philosophy

The TPM is built upon a very simple timer (HCS08 Timer PWM Module – TPM) used for many years on NXP'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.

33.1.2 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

33.1.3 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.

33.1.4 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.

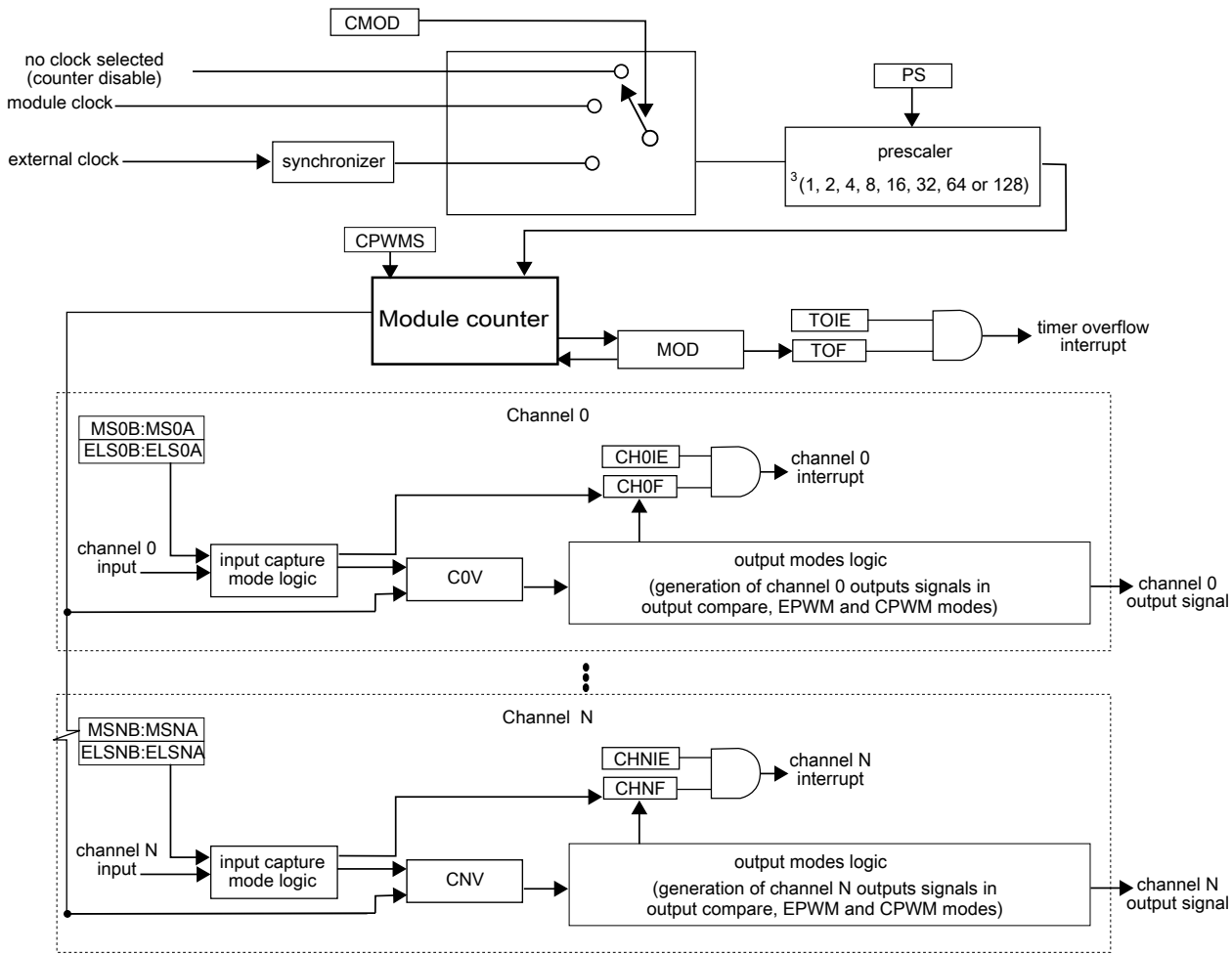


Figure 33-1. TPM block diagram

33.2 TPM Signal Descriptions

Table 33-1 shows the user-accessible signals for the TPM.

Table 33-1. 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

33.2.1 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.

33.2.2 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.

33.3 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.

NOTE

Not all the registers are available for this device, See chip-specific sections for details.

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	33.3.1/793

Table continues on the next page...

TPM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8004	Counter (TPM0_CNT)	32	R/W	0000_0000h	33.3.2/794
4003_8008	Modulo (TPM0_MOD)	32	R/W	0000_FFFFh	33.3.3/795
4003_800C	Channel (n) Status and Control (TPM0_C0SC)	32	R/W	0000_0000h	33.3.4/796
4003_8010	Channel (n) Value (TPM0_C0V)	32	R/W	0000_0000h	33.3.5/798
4003_8014	Channel (n) Status and Control (TPM0_C1SC)	32	R/W	0000_0000h	33.3.4/796
4003_8018	Channel (n) Value (TPM0_C1V)	32	R/W	0000_0000h	33.3.5/798
4003_801C	Channel (n) Status and Control (TPM0_C2SC)	32	R/W	0000_0000h	33.3.4/796
4003_8020	Channel (n) Value (TPM0_C2V)	32	R/W	0000_0000h	33.3.5/798
4003_8024	Channel (n) Status and Control (TPM0_C3SC)	32	R/W	0000_0000h	33.3.4/796
4003_8028	Channel (n) Value (TPM0_C3V)	32	R/W	0000_0000h	33.3.5/798
4003_8050	Capture and Compare Status (TPM0_STATUS)	32	R/W	0000_0000h	33.3.6/798
4003_8064	Combine Channel Register (TPM0_COMBINE)	32	R/W	0000_0000h	33.3.7/800
4003_8070	Channel Polarity (TPM0_POL)	32	R/W	0000_0000h	33.3.8/801
4003_8078	Filter Control (TPM0_FILTER)	32	R/W	0000_0000h	33.3.9/802
4003_8080	Quadrature Decoder Control and Status (TPM0_QDCTRL)	32	R/W	0000_0000h	33.3.10/ 803
4003_8084	Configuration (TPM0_CONF)	32	R/W	0000_0000h	33.3.11/ 804
4003_9000	Status and Control (TPM1_SC)	32	R/W	0000_0000h	33.3.1/793
4003_9004	Counter (TPM1_CNT)	32	R/W	0000_0000h	33.3.2/794
4003_9008	Modulo (TPM1_MOD)	32	R/W	0000_FFFFh	33.3.3/795
4003_900C	Channel (n) Status and Control (TPM1_C0SC)	32	R/W	0000_0000h	33.3.4/796
4003_9010	Channel (n) Value (TPM1_C0V)	32	R/W	0000_0000h	33.3.5/798
4003_9014	Channel (n) Status and Control (TPM1_C1SC)	32	R/W	0000_0000h	33.3.4/796
4003_9018	Channel (n) Value (TPM1_C1V)	32	R/W	0000_0000h	33.3.5/798
4003_901C	Channel (n) Status and Control (TPM1_C2SC)	32	R/W	0000_0000h	33.3.4/796
4003_9020	Channel (n) Value (TPM1_C2V)	32	R/W	0000_0000h	33.3.5/798
4003_9024	Channel (n) Status and Control (TPM1_C3SC)	32	R/W	0000_0000h	33.3.4/796
4003_9028	Channel (n) Value (TPM1_C3V)	32	R/W	0000_0000h	33.3.5/798
4003_9050	Capture and Compare Status (TPM1_STATUS)	32	R/W	0000_0000h	33.3.6/798
4003_9064	Combine Channel Register (TPM1_COMBINE)	32	R/W	0000_0000h	33.3.7/800
4003_9070	Channel Polarity (TPM1_POL)	32	R/W	0000_0000h	33.3.8/801
4003_9078	Filter Control (TPM1_FILTER)	32	R/W	0000_0000h	33.3.9/802
4003_9080	Quadrature Decoder Control and Status (TPM1_QDCTRL)	32	R/W	0000_0000h	33.3.10/ 803
4003_9084	Configuration (TPM1_CONF)	32	R/W	0000_0000h	33.3.11/ 804
4003_A000	Status and Control (TPM2_SC)	32	R/W	0000_0000h	33.3.1/793
4003_A004	Counter (TPM2_CNT)	32	R/W	0000_0000h	33.3.2/794

Table continues on the next page...

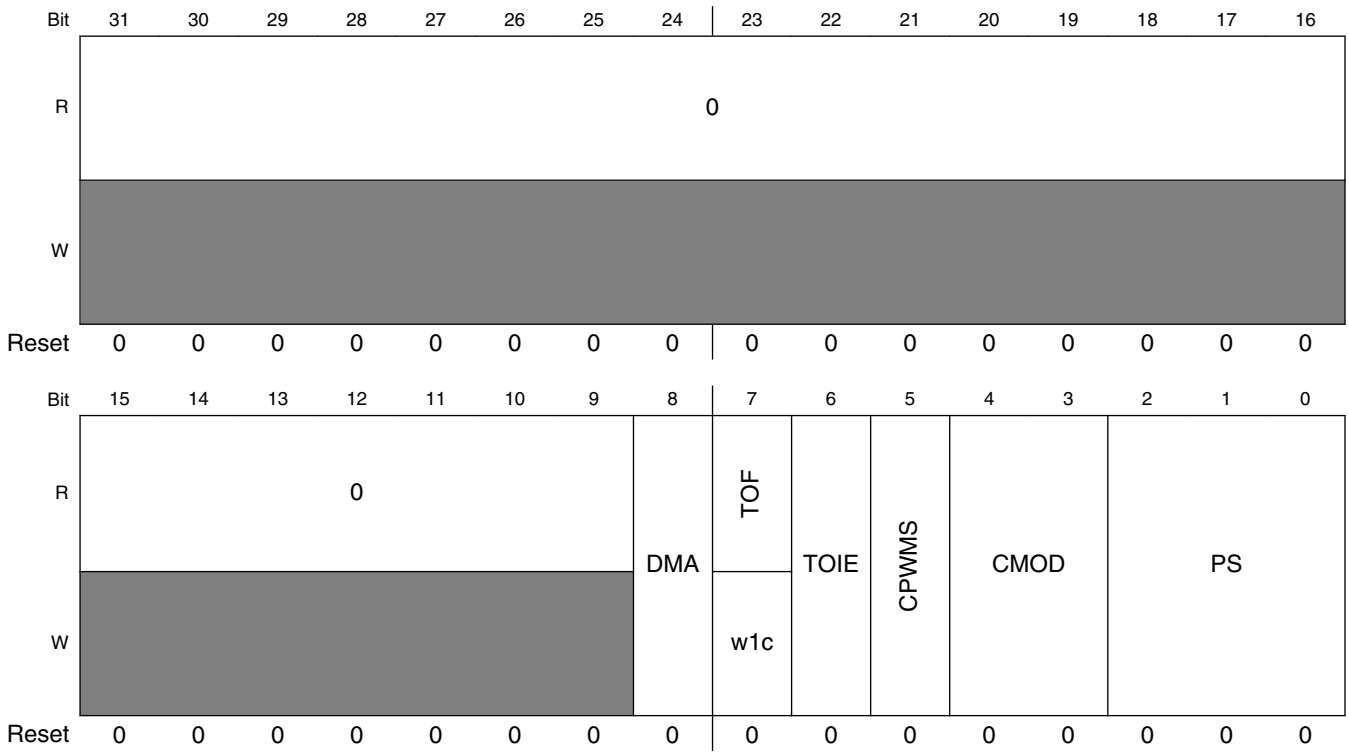
TPM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_A008	Modulo (TPM2_MOD)	32	R/W	0000_FFFFh	33.3.3/795
4003_A00C	Channel (n) Status and Control (TPM2_C0SC)	32	R/W	0000_0000h	33.3.4/796
4003_A010	Channel (n) Value (TPM2_C0V)	32	R/W	0000_0000h	33.3.5/798
4003_A014	Channel (n) Status and Control (TPM2_C1SC)	32	R/W	0000_0000h	33.3.4/796
4003_A018	Channel (n) Value (TPM2_C1V)	32	R/W	0000_0000h	33.3.5/798
4003_A01C	Channel (n) Status and Control (TPM2_C2SC)	32	R/W	0000_0000h	33.3.4/796
4003_A020	Channel (n) Value (TPM2_C2V)	32	R/W	0000_0000h	33.3.5/798
4003_A024	Channel (n) Status and Control (TPM2_C3SC)	32	R/W	0000_0000h	33.3.4/796
4003_A028	Channel (n) Value (TPM2_C3V)	32	R/W	0000_0000h	33.3.5/798
4003_A050	Capture and Compare Status (TPM2_STATUS)	32	R/W	0000_0000h	33.3.6/798
4003_A064	Combine Channel Register (TPM2_COMBINE)	32	R/W	0000_0000h	33.3.7/800
4003_A070	Channel Polarity (TPM2_POL)	32	R/W	0000_0000h	33.3.8/801
4003_A078	Filter Control (TPM2_FILTER)	32	R/W	0000_0000h	33.3.9/802
4003_A080	Quadrature Decoder Control and Status (TPM2_QDCTRL)	32	R/W	0000_0000h	33.3.10/ 803
4003_A084	Configuration (TPM2_CONF)	32	R/W	0000_0000h	33.3.11/ 804

33.3.1 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



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.

Table continues on the next page...

TPMx_SC field descriptions (continued)

Field	Description
	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 32 110 Divide by 64 111 Divide by 128

33.3.2 Counter (TPMx_CNT)

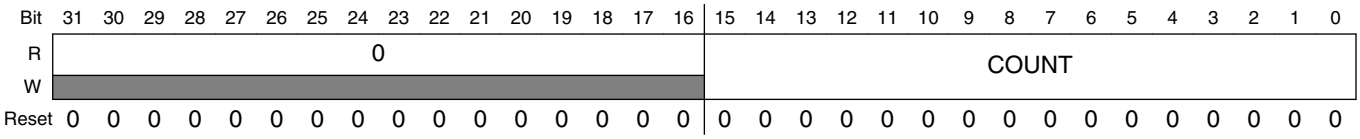
The CNT register contains the TPM counter value.

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



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

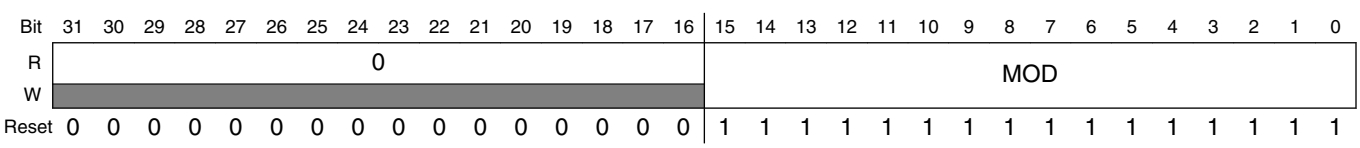
33.3.3 Modulo (TPMx_MOD)

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



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.

33.3.4 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 33-2. 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
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 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.

33.3.5 Channel (n) Value (TPMx_C*n*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_C*n*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.

33.3.6 Capture and Compare Status (TPMx_STATUS)

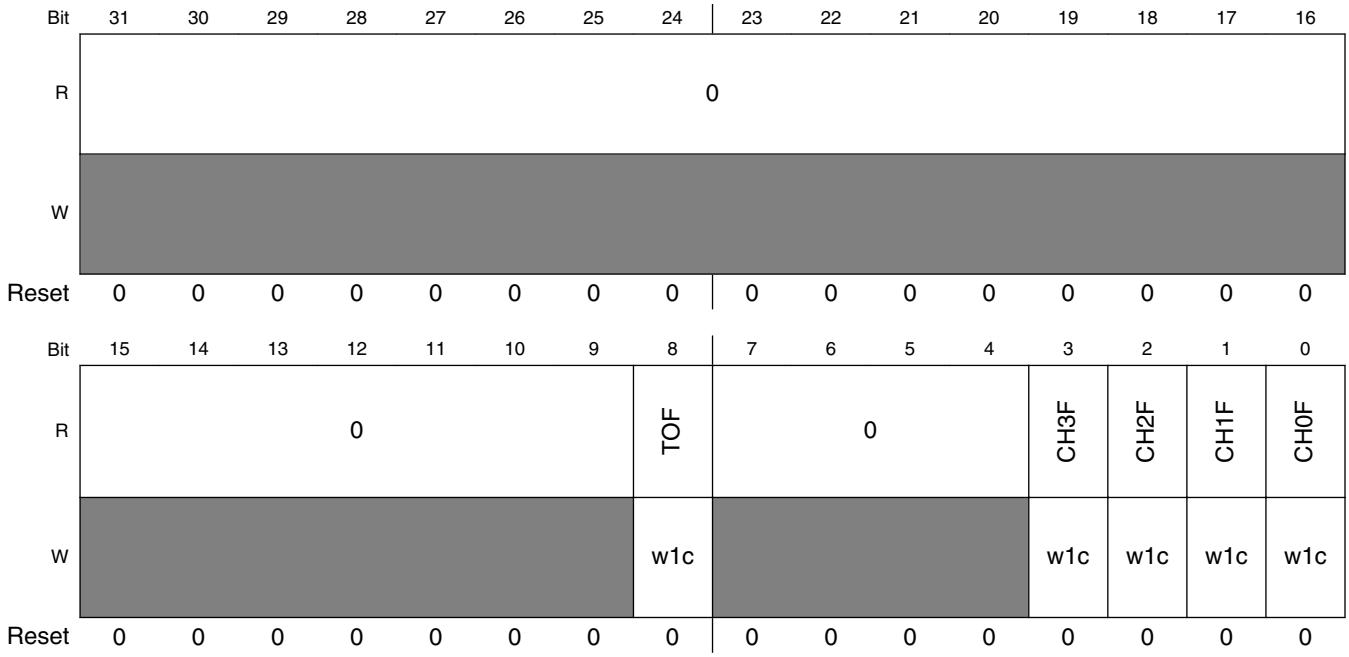
The STATUS register contains a copy of the status flag, CnSC[CHnF] for each TPM channel, as well as SC[TOF], for software convenience.

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



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 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.

Table continues on the next page...

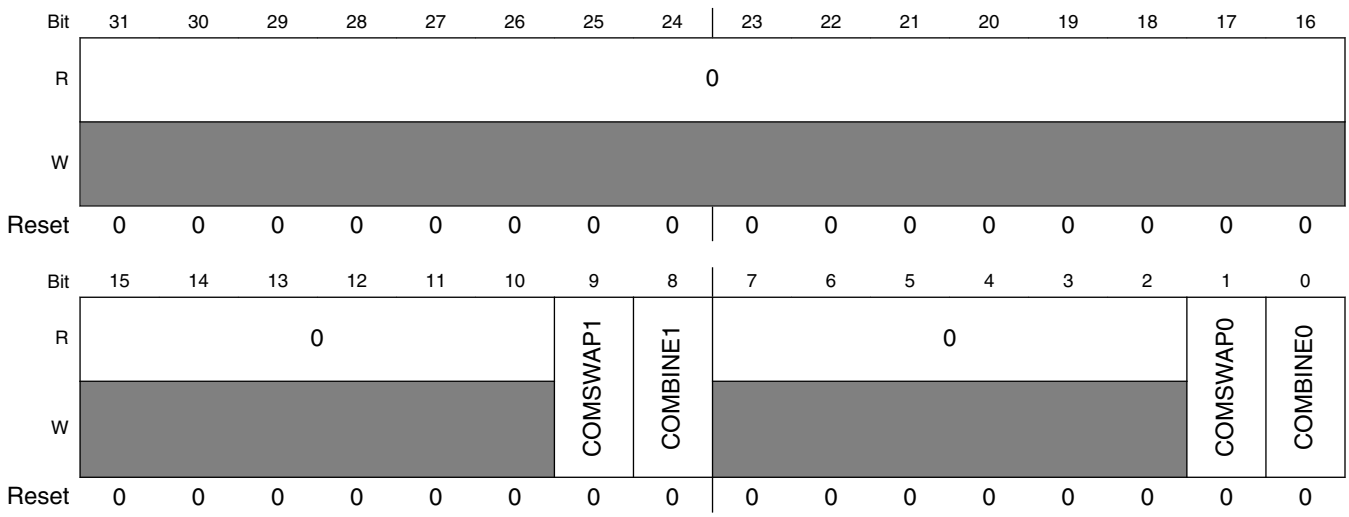
TPMx_STATUS field descriptions (continued)

Field	Description
0 CH0F	Channel 0 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.

33.3.7 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



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

Table continues on the next page...

TPMx_COMBINE field descriptions (continued)

Field	Description
	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 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.

33.3.8 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

Table continues on the next page...

TPMx_POL field descriptions (continued)

Field	Description
	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.
0 POL0	Channel 0 Polarity 0 The channel polarity is active high. 1 The channel polarity is active low.

33.3.9 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

Table continues on the next page...

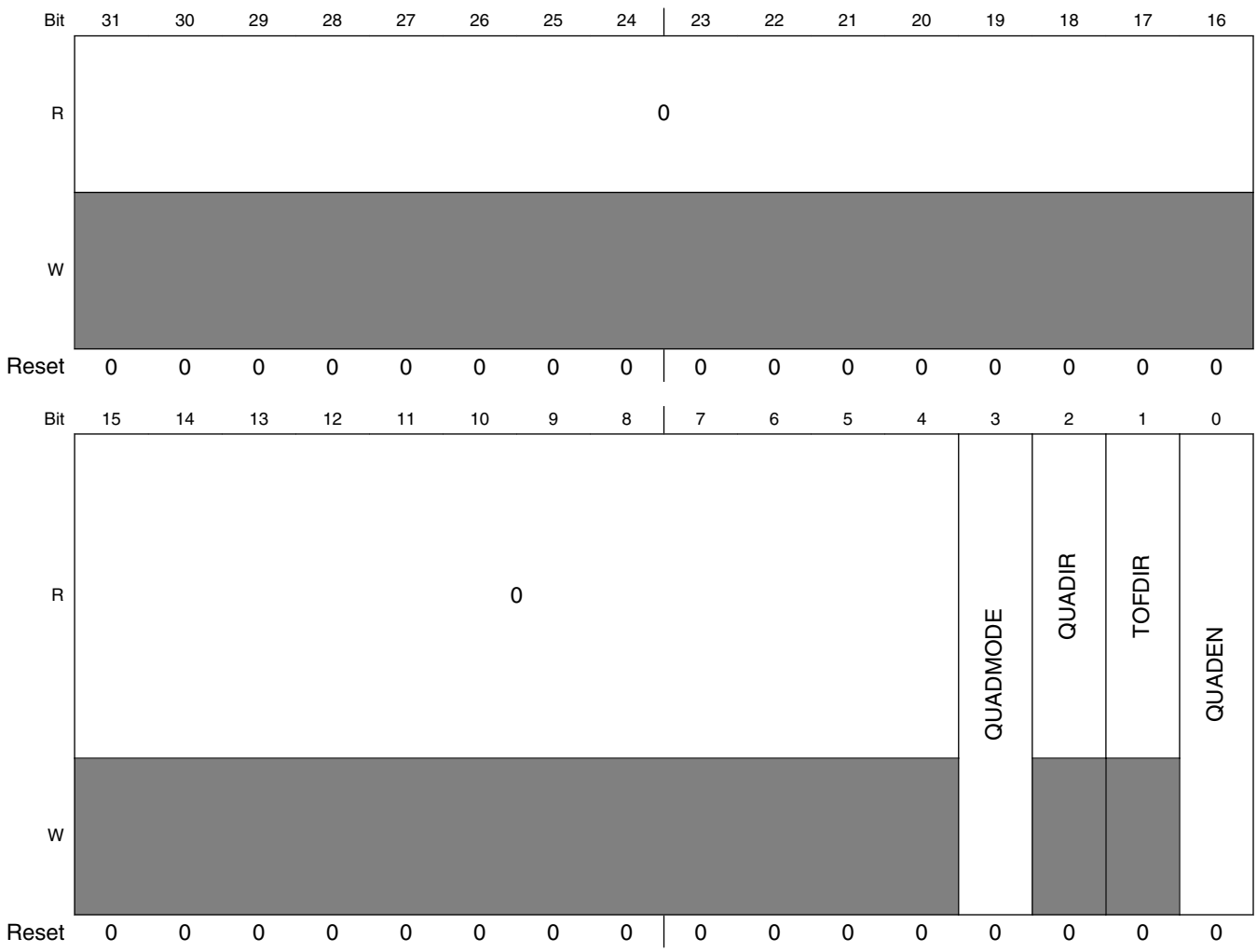
TPMx_FILTER field descriptions (continued)

Field	Description
	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.

33.3.10 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



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.

Table continues on the next page...

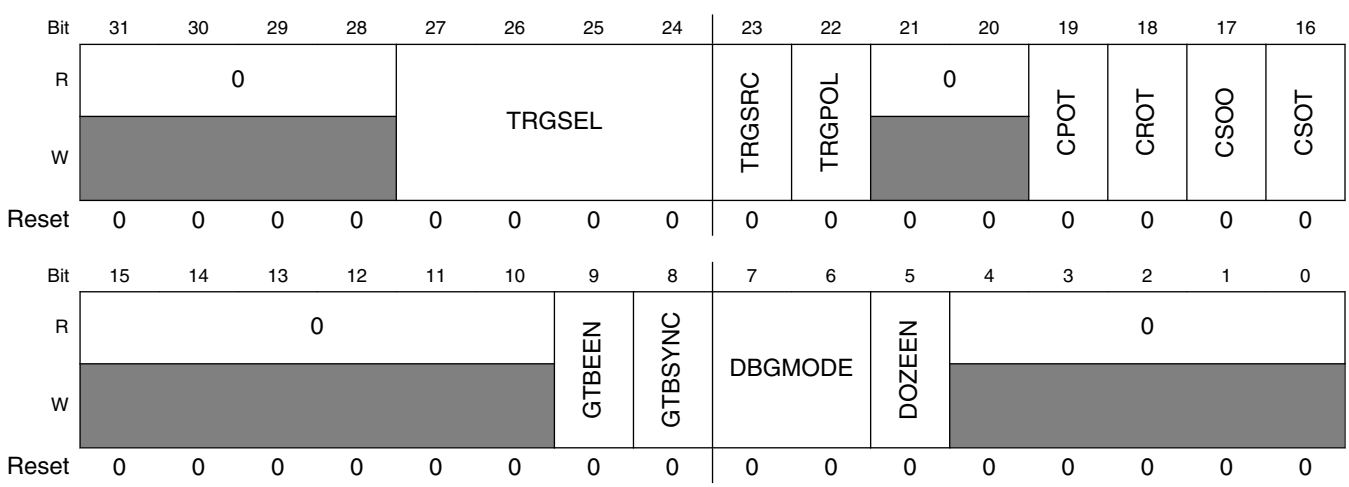
TPMx_QDCTRL field descriptions (continued)

Field	Description
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 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.

33.3.11 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



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.
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.

Table continues on the next page...

TPMx_CONF field descriptions (continued)

Field	Description
18 CROT	Counter Reload On Trigger 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

Table continues on the next page...

TPMx_CONF field descriptions (continued)

Field	Description
	Configures the TPM behavior in debug mode. All other configurations are reserved. 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.

33.4 Functional description

The following sections describe the TPM features.

33.4.1 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.

33.4.1.1 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.

33.4.2 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.

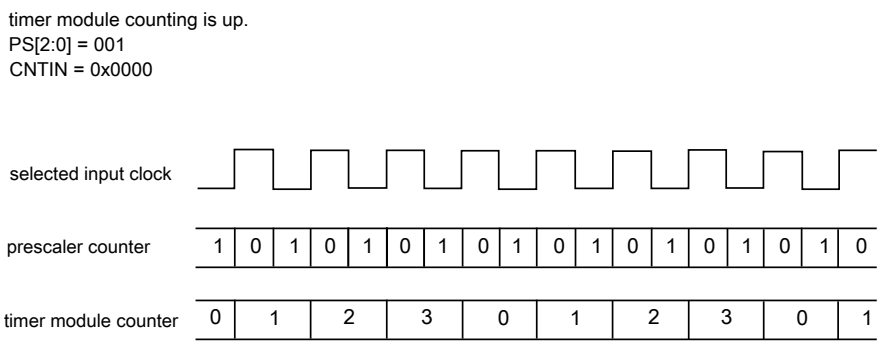


Figure 33-2. Example of the Prescaler Counter

33.4.3 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)

33.4.3.1 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.

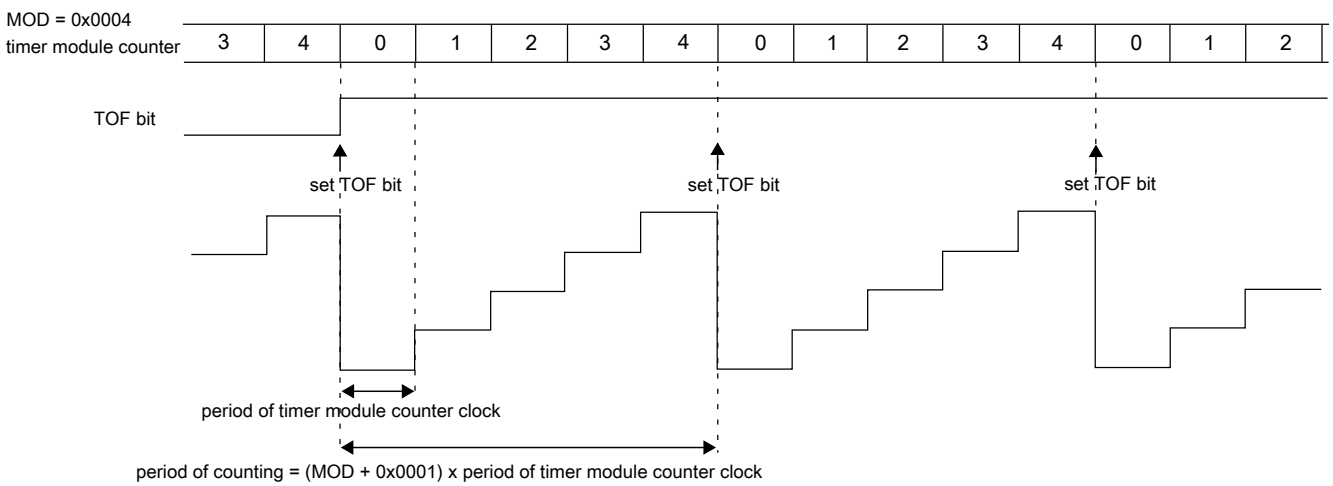


Figure 33-3. Example of TPM Up Counting

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.

33.4.3.2 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).

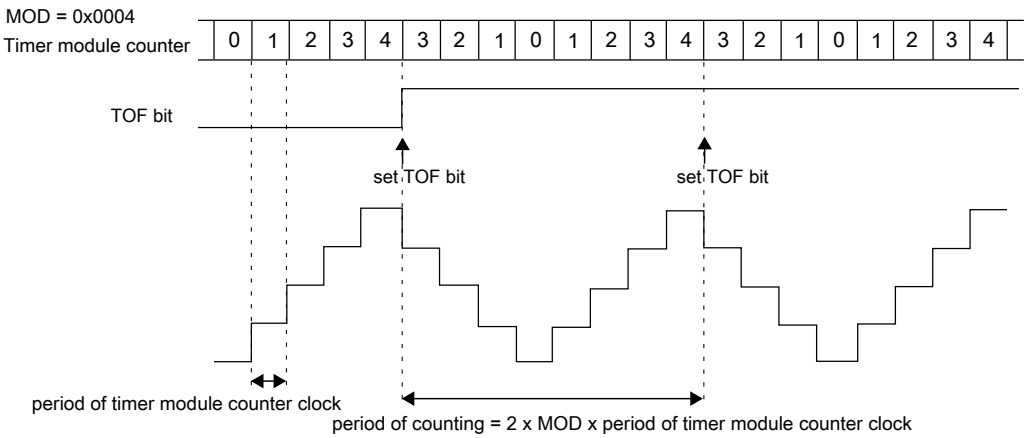


Figure 33-4. Example of up-down counting

33.4.3.3 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).

33.4.3.4 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).

33.4.3.5 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.

33.4.4 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.

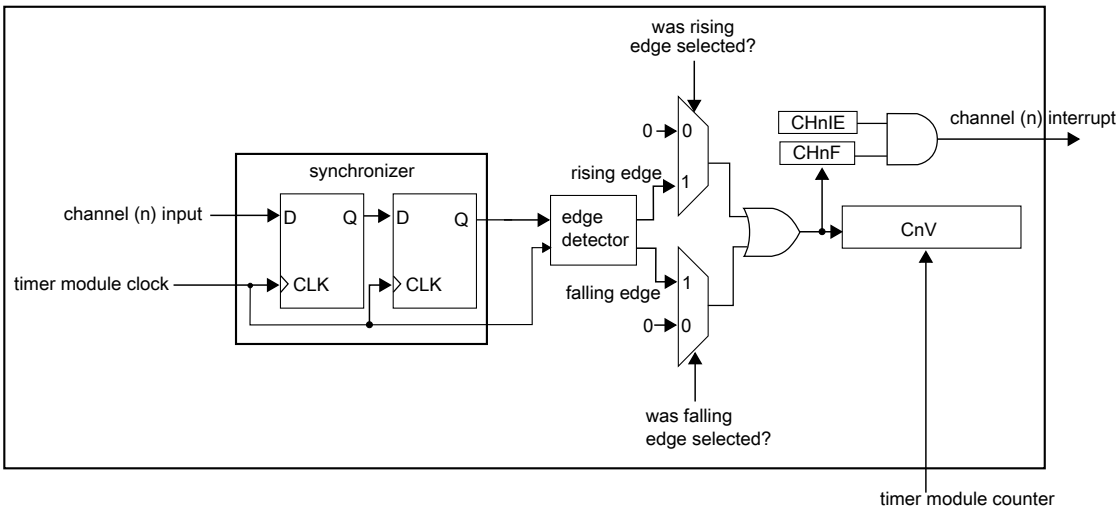


Figure 33-5. Input capture mode

The CHnF bit is set on the third rising edge of the counter clock after a valid edge occurs on the channel input.

33.4.5 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).

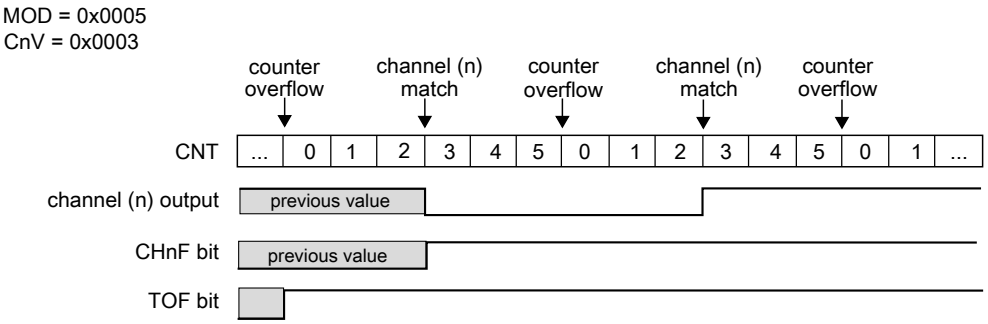


Figure 33-6. Example of the output compare mode when the match toggles the channel output

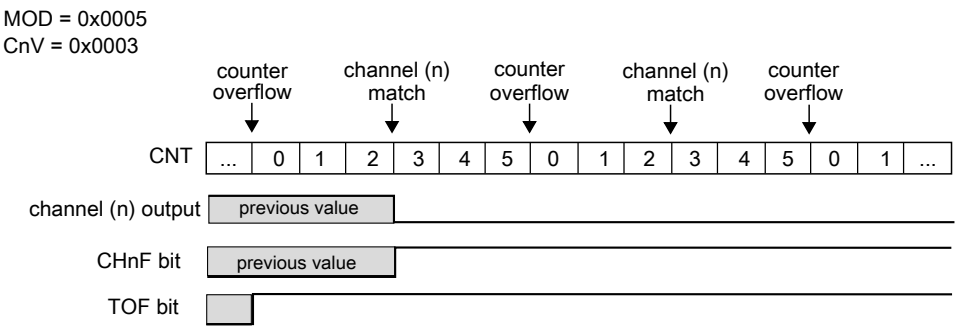


Figure 33-7. Example of the output compare mode when the match clears the channel output

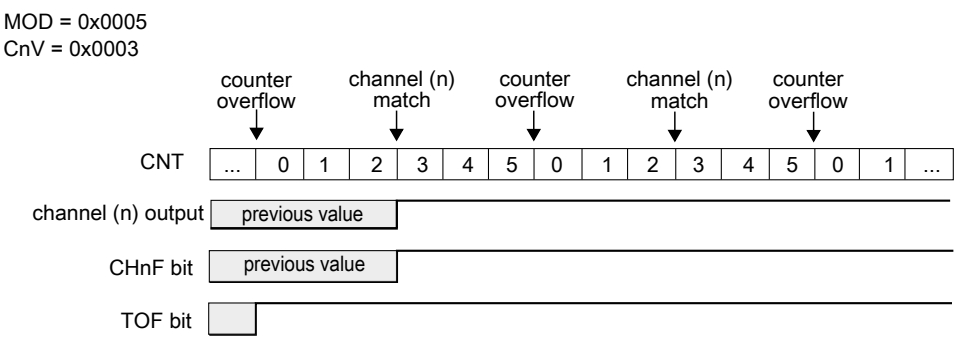


Figure 33-8. Example of the output compare mode when the match sets the channel output

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.

33.4.6 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.

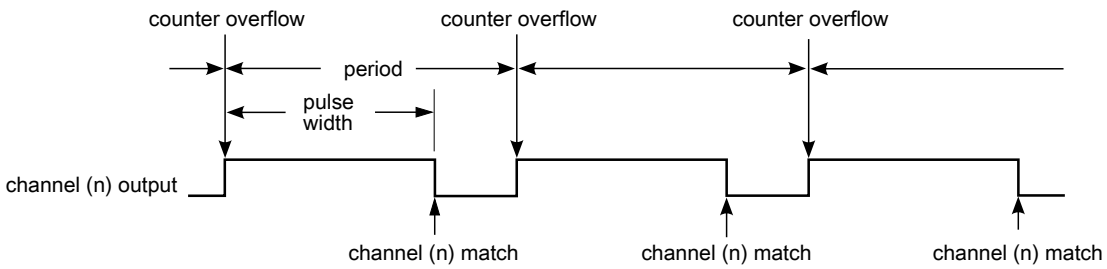


Figure 33-9. EPWM period and pulse width with ELSnB:ELSnA = 1:0

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).

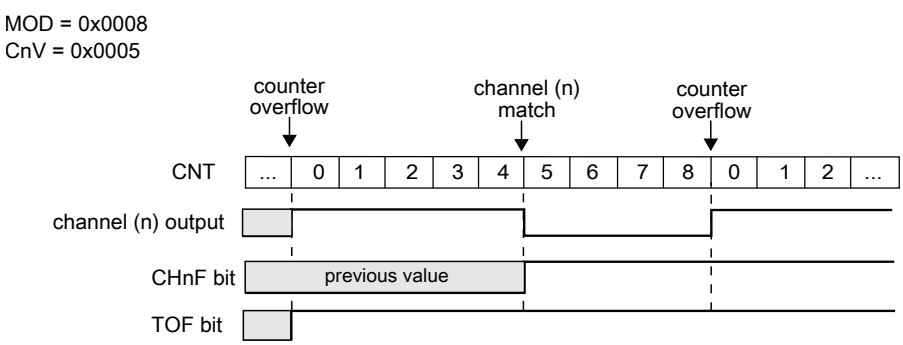


Figure 33-10. EPWM signal with ELSnB:ELSnA = 1:0

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).

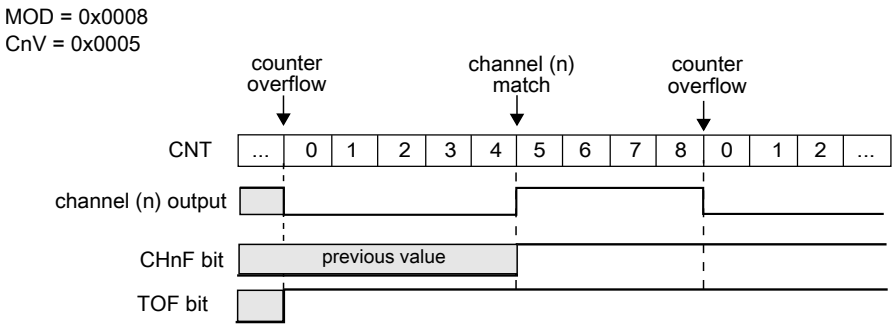


Figure 33-11. EPWM signal with ELSnB:ELSnA = X:1

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.

33.4.7 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).

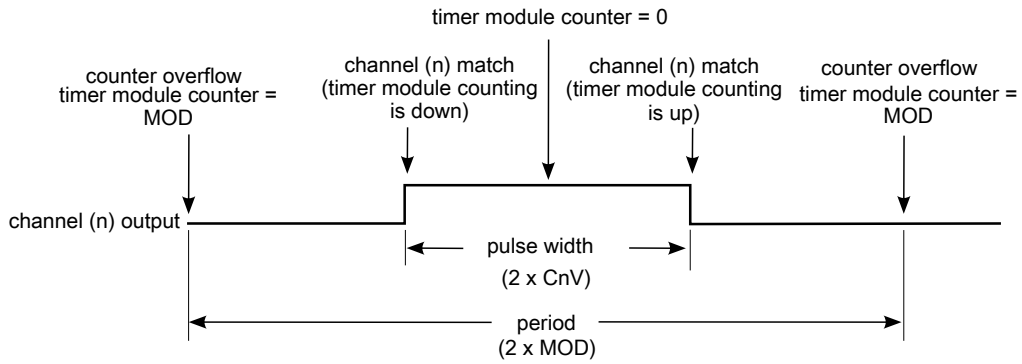


Figure 33-12. CPWM period and pulse width with ELSnB:ELSnA = 1:0

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).

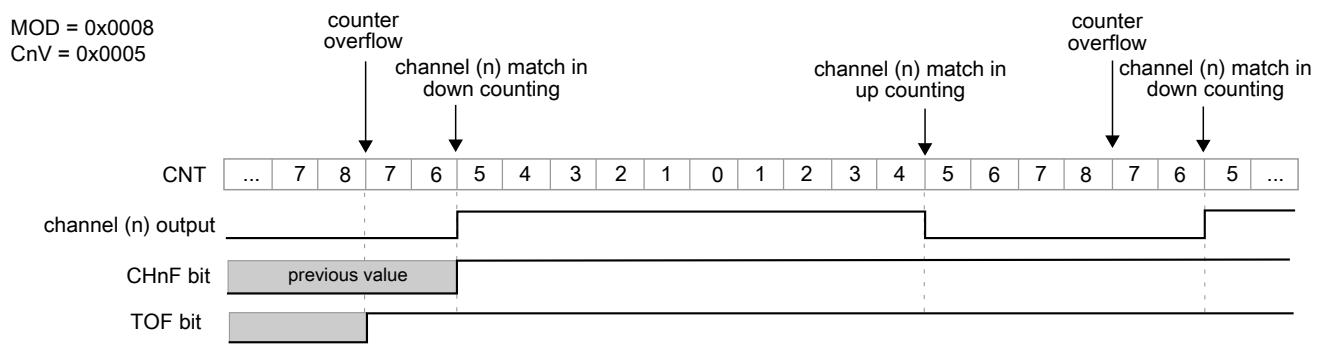


Figure 33-13. CPWM signal with ELSnB:ELSnA = 1:0

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).

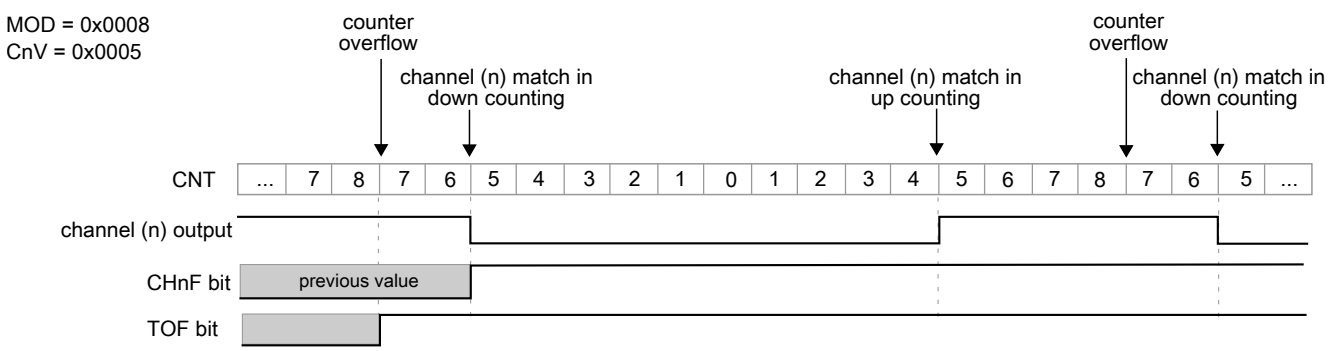


Figure 33-14. CPWM signal with ELSnB:ELSnA = X:1

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.

33.4.8 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.

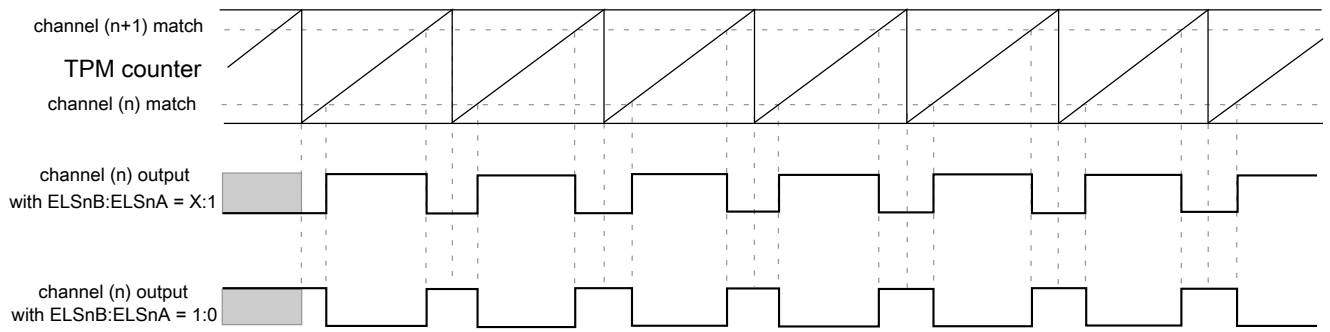


Figure 33-15. Combine mode

The following figures illustrate the PWM signals generation using Combine mode.

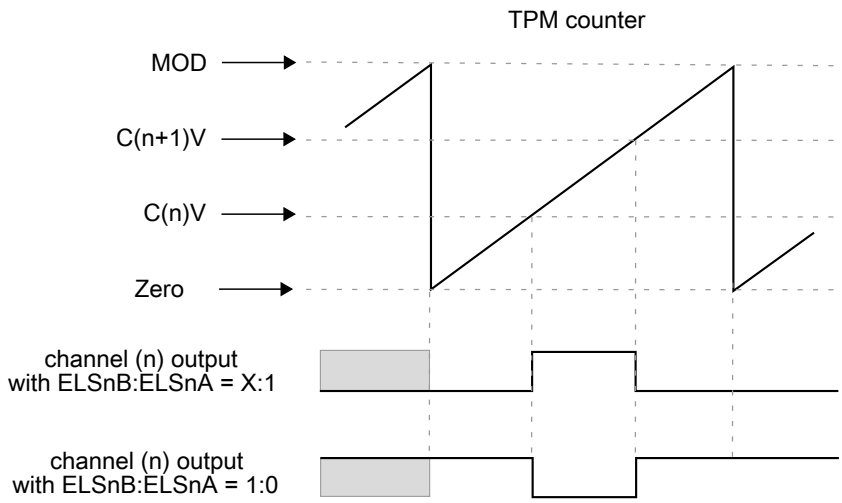


Figure 33-16. Channel (n) output if (C(n)V < MOD) and (C(n+1)V < MOD) and (C(n)V < C(n+1)V)

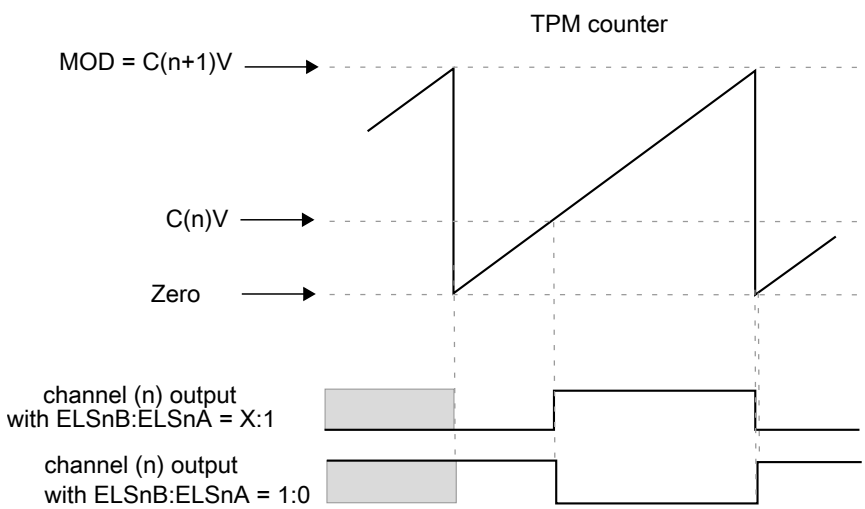


Figure 33-17. Channel (n) output if (C(n)V < MOD) and (C(n+1)V = MOD)

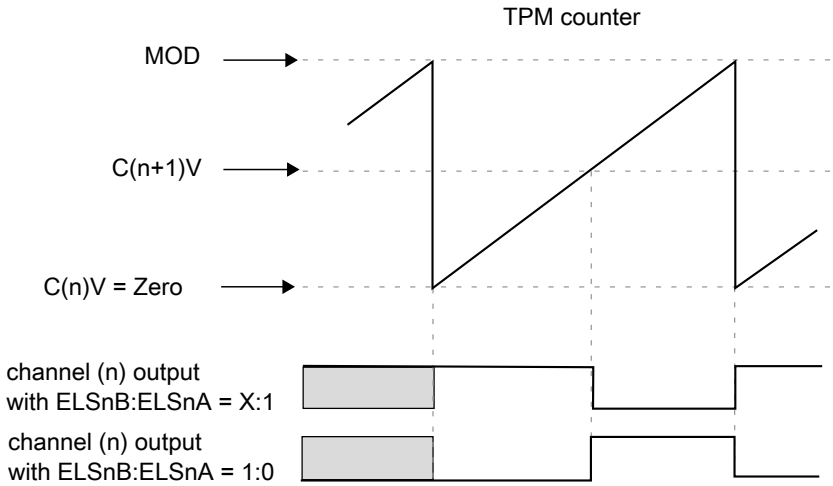


Figure 33-18. Channel (n) output if (C(n)V = zero) and (C(n+1)V < MOD)

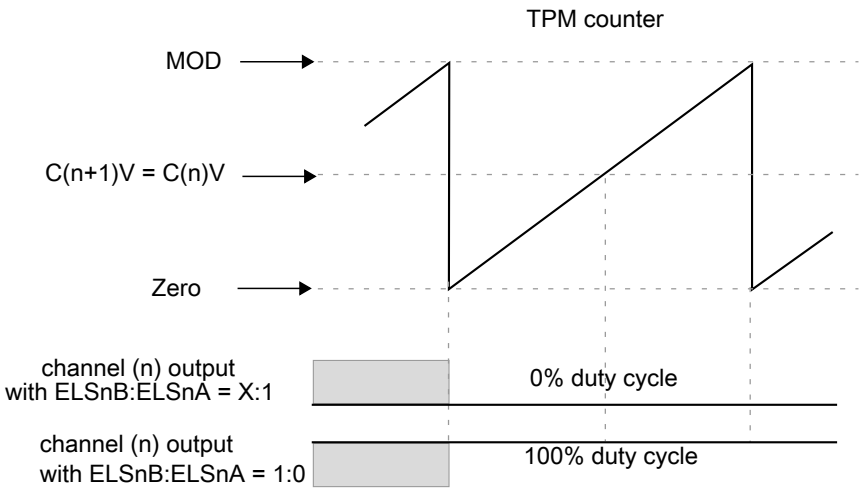


Figure 33-19. Channel (n) output if (C(n)V < MOD) and (C(n+1)V < MOD) and (C(n)V = C(n +1)V)

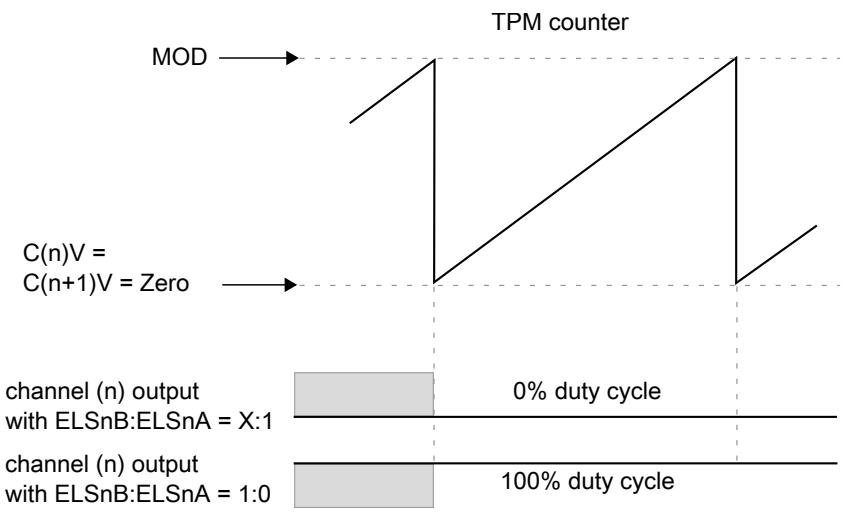


Figure 33-20. Channel (n) output if (C(n)V = C(n+1)V = zero)

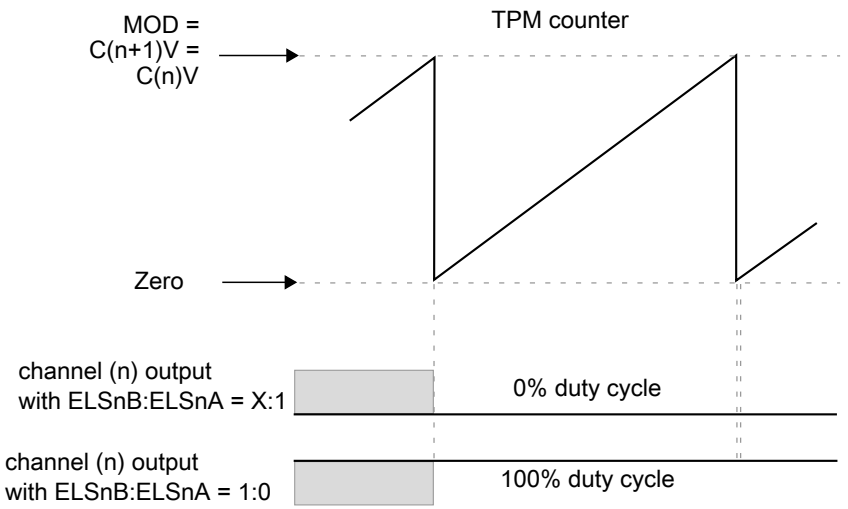


Figure 33-21. Channel (n) output if (C(n)V = C(n+1)V = MOD)

33.4.9 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.

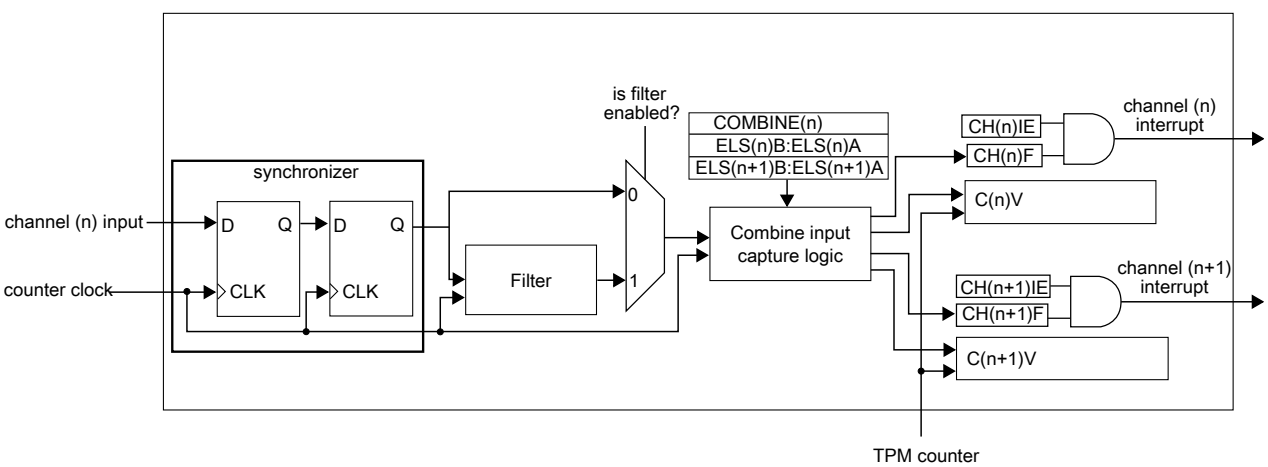


Figure 33-22. Combine Input Capture mode block diagram

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.

33.4.10 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.

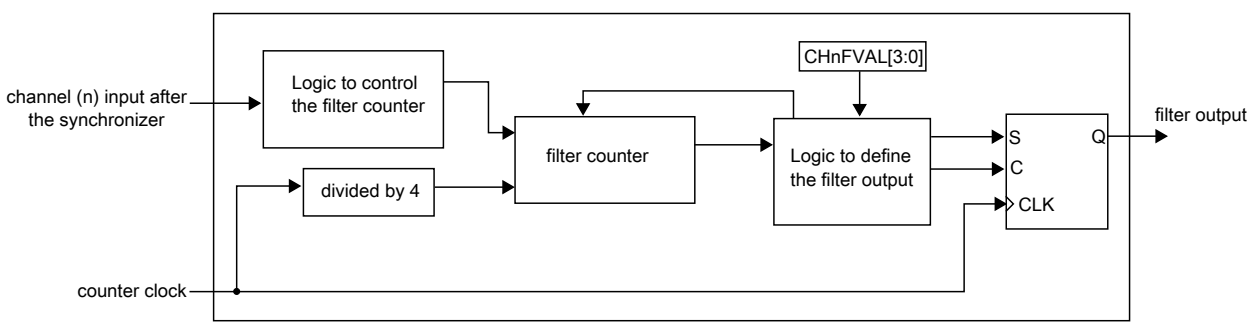


Figure 33-23. Channel input filter

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.

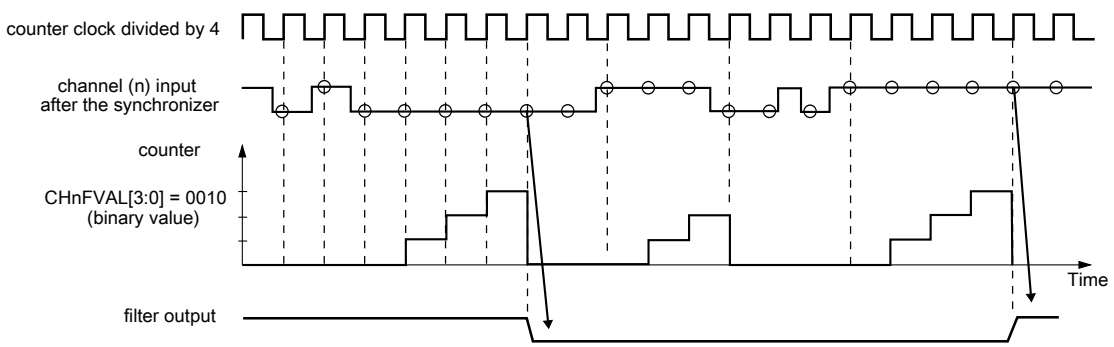


Figure 33-24. Channel input filter example

33.4.11 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.

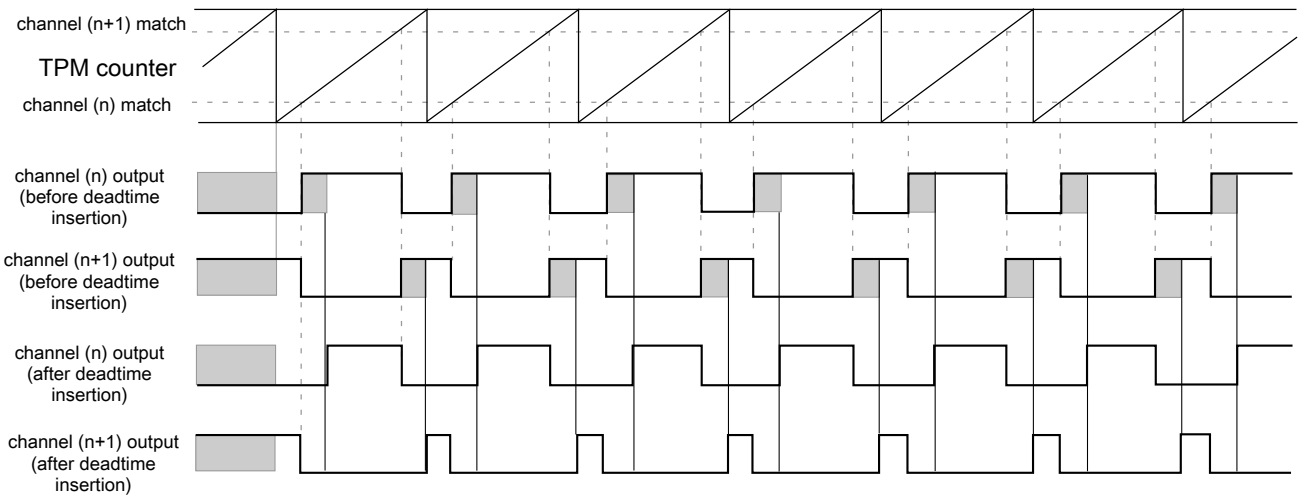


Figure 33-25. Deadtime insertion with ELSnB:ELSnA = X:1, POL(n) = 0, and POL(n+1) = 1

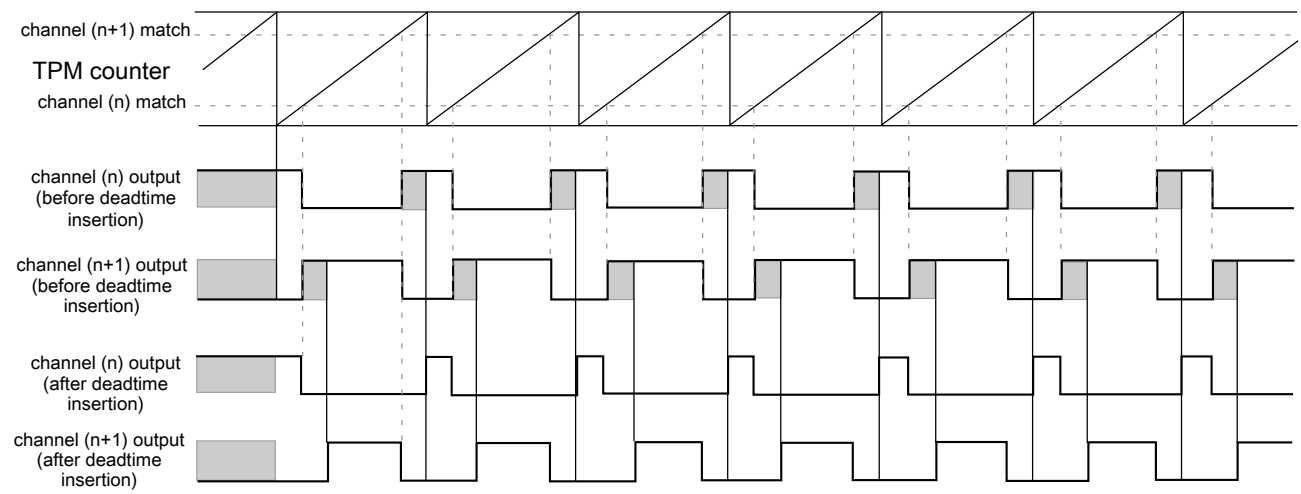


Figure 33-26. Deadtime insertion with ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 1

33.4.12 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.

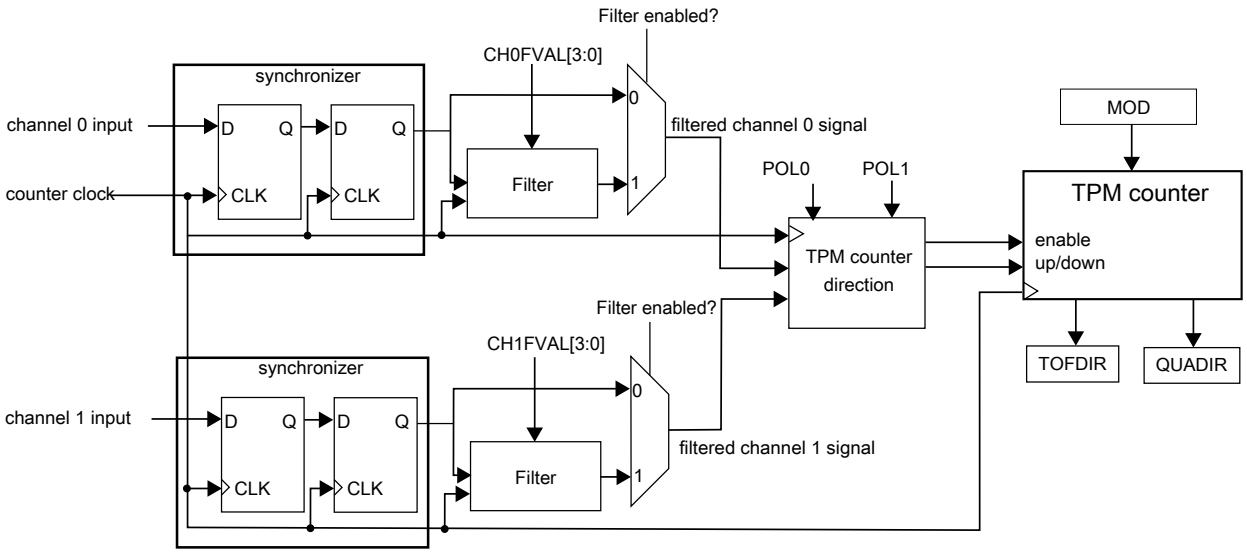


Figure 33-27. Quadrature Decoder block diagram

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.

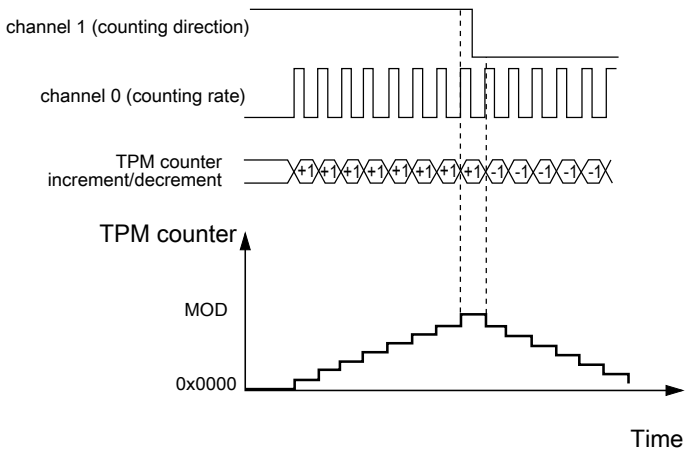


Figure 33-28. Quadrature Decoder – Count and Direction Encoding mode

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.

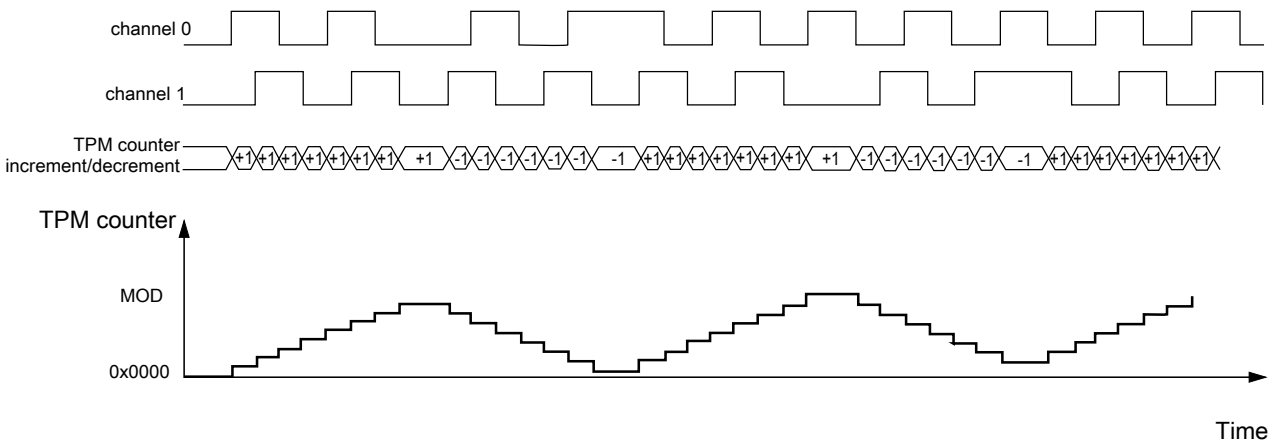


Figure 33-29. Quadrature Decoder – Phase Encoding mode

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.

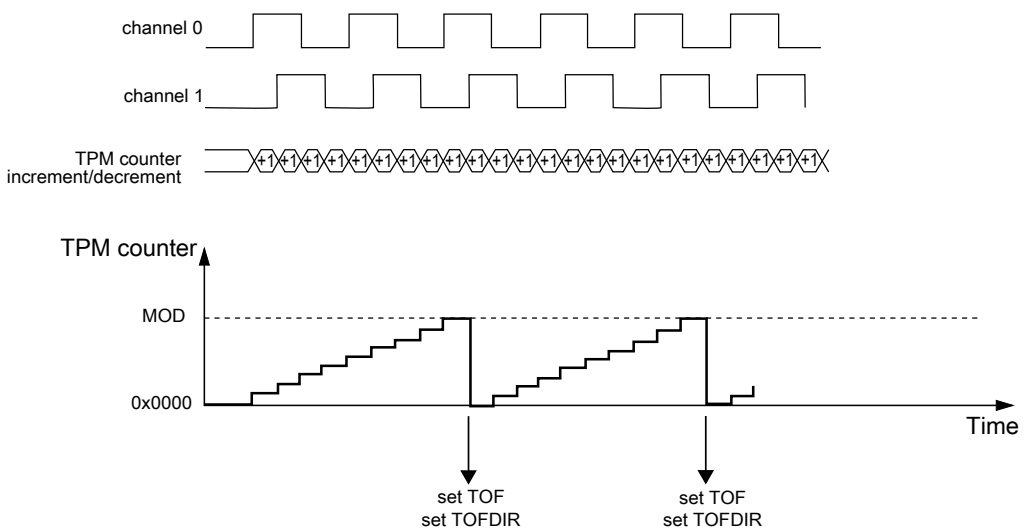


Figure 33-30. TPM Counter overflow in up counting for Quadrature Decoder mode

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.

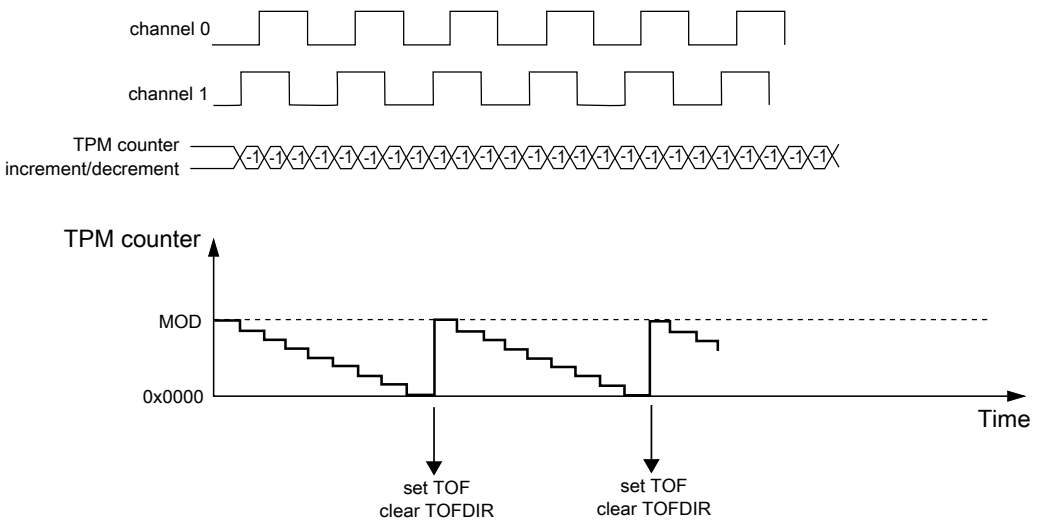


Figure 33-31. TPM counter overflow in down counting for Quadrature Decoder mode

33.4.13 Registers Updated from Write Buffers

33.4.13.1 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).

33.4.13.2 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).

33.4.14 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 33-3. 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 33-4. 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.

33.4.15 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.

33.4.16 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).

33.4.17 TPM Interrupts

This section describes TPM interrupts.

33.4.17.1 Timer Overflow Interrupt

The timer overflow interrupt is generated when (TOIE = 1) and (TOF = 1).

33.4.17.2 Channel (n) Interrupt

The channel (n) interrupt is generated when (CHnIE = 1) and (CHnF = 1).

Chapter 34
Periodic interrupt timer (PIT)

34.1 Introduction

The PIT module is an array of timers that can be used to raise interrupts and triggers.

34.1.1 Block diagram

The following figure shows the block diagram of PIT module.

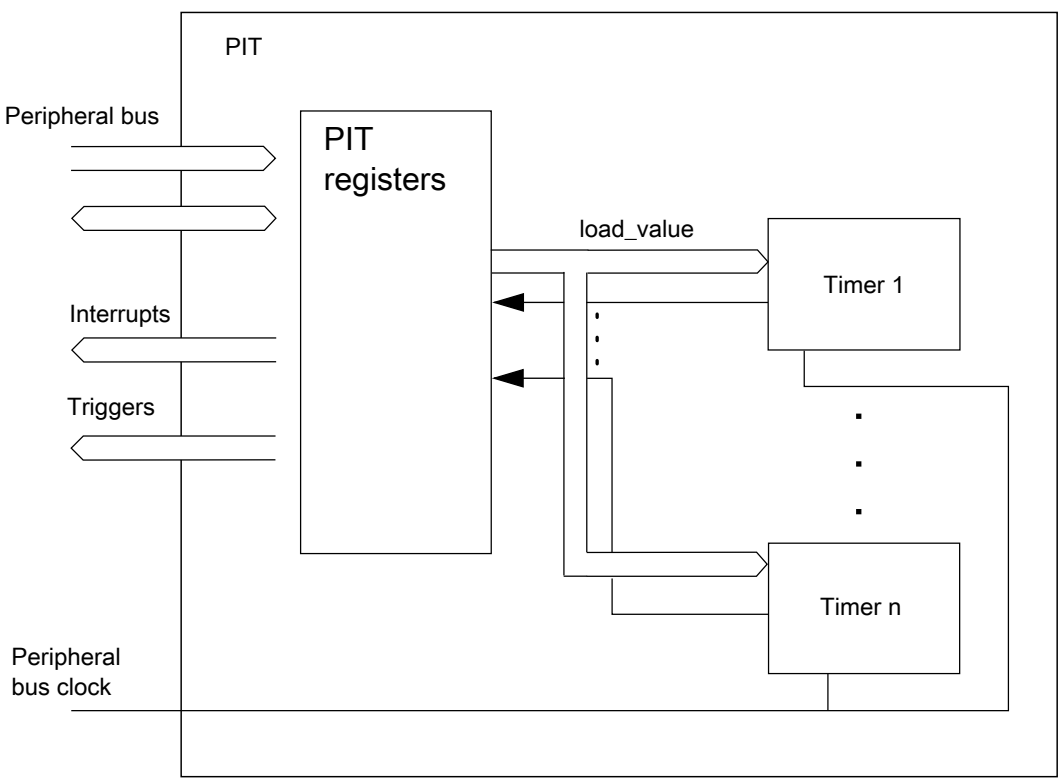


Figure 34-1. Block diagram of PIT

34.1.2 Features

The key features of the module are these:

- Ability of timers to generate trigger pulses
- Ability of timers to generate interrupts
- Maskable interrupts
- Independent timeout periods for each timer

34.2 Modes of operation

This subsection briefly describes all operating modes supported by PIT.

- Run mode

  All functional parts of the PIT are running during normal Run mode.

34.3 PIT register descriptions

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.

34.3.1 PIT Memory map

PIT base address: 4003_7000h

Offset	Register	Width (In bits)	Access	Reset value
0h	PIT Module Control Register (MCR)	32	RW	0000_0002h
E0h	PIT Upper Lifetime Timer Register (LTMR64H)	32	RO	0000_0000h
E4h	PIT Lower Lifetime Timer Register (LTMR64L)	32	RO	0000_0000h
100h	Timer Load Value Register (LDVAL0)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
104h	Current Timer Value Register (CVAL0)	32	RO	0000_0000h
108h	Timer Control Register (TCTRL0)	32	RW	0000_0000h
10Ch	Timer Flag Register (TFLG0)	32	W1C	0000_0000h
110h	Timer Load Value Register (LDVAL1)	32	RW	0000_0000h
114h	Current Timer Value Register (CVAL1)	32	RO	0000_0000h
118h	Timer Control Register (TCTRL1)	32	RW	0000_0000h
11Ch	Timer Flag Register (TFLG1)	32	W1C	0000_0000h

34.3.2 PIT Module Control Register (MCR)

34.3.2.1 Offset

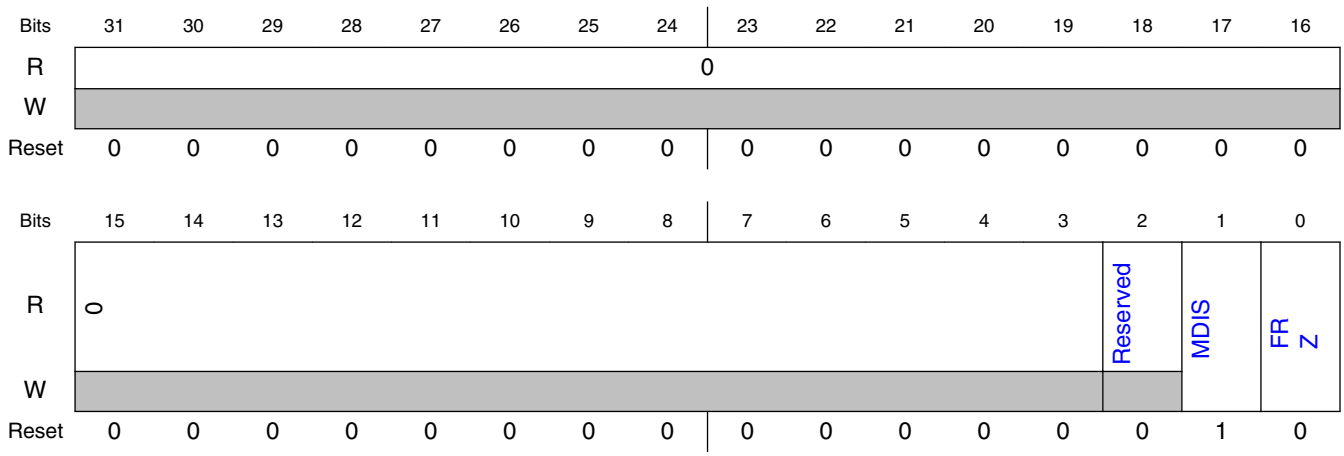
Register	Offset
MCR	0h

34.3.2.2 Function

This register enables or disables the PIT timer clocks and controls the timers when the PIT enters the Debug mode.

Access: User read/write

34.3.2.3 Diagram



34.3.2.4 Fields

Field	Function
31-3 —	Reserved
2 —	Reserved
1 MDIS	Module Disable for PIT Disables the standard timers. The field must be enabled before any other setup is done. 0b - Clock for standard PIT timers is enabled. 1b - Clock for standard PIT timers is disabled.
0 FRZ	Freeze Allows the timers to be stopped when the device enters the Debug mode. 0b - Timers continue to run in Debug mode. 1b - Timers are stopped in Debug mode.

34.3.3 PIT Upper Lifetime Timer Register (LTMR64H)

34.3.3.1 Offset

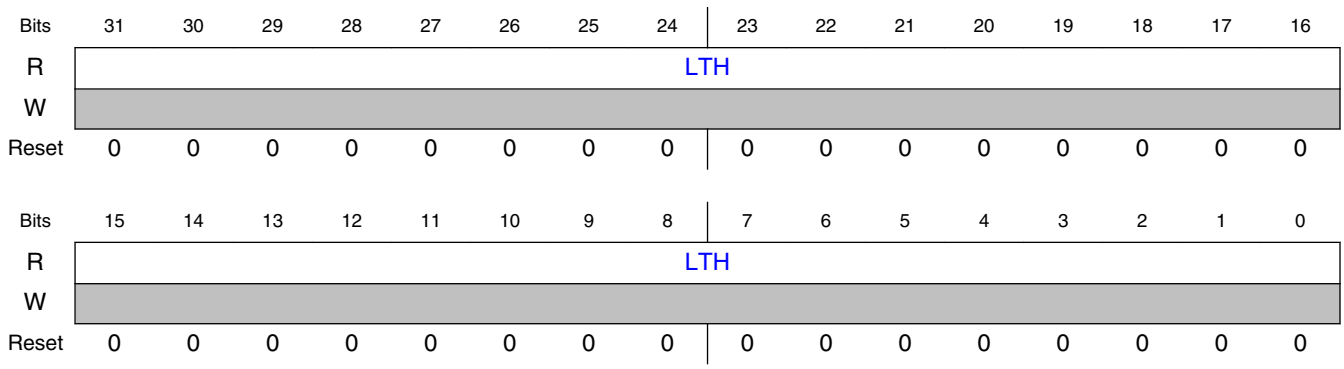
Register	Offset
LTMR64H	E0h

34.3.3.2 Function

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit lifetimer.

Access: User read only

34.3.3.3 Diagram



34.3.3.4 Fields

Field	Function
31-0 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.

34.3.4 PIT Lower Lifetime Timer Register (LTMR64L)

34.3.4.1 Offset

Register	Offset
LTMR64L	E4h

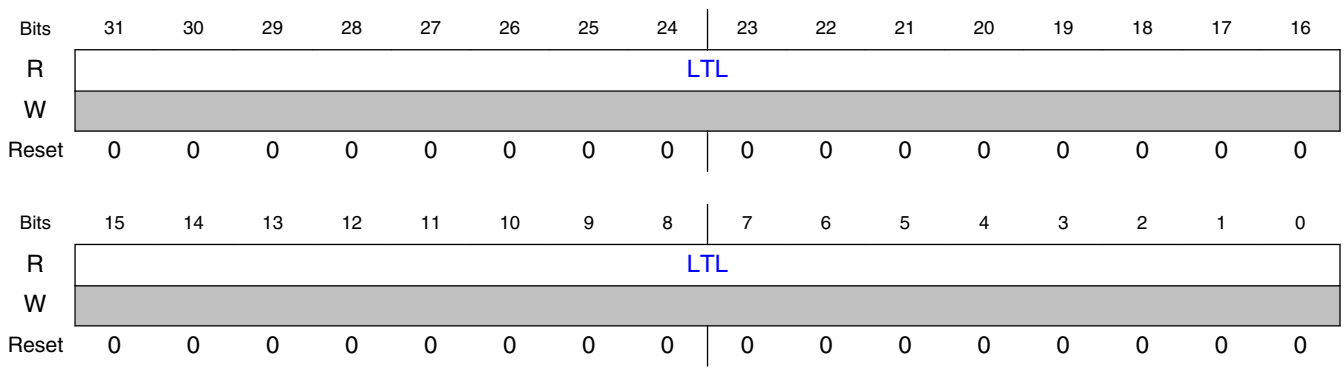
34.3.4.2 Function

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. The value for the LTMR64H register is set to CVAL1 at the time of the first access and the value of the LTMR64L register is set to CVAL0 at first access. Therefore, the application is not affected by the carry-over effects of the running counter.

Access: User read only

34.3.4.3 Diagram



34.3.4.4 Fields

Field	Function
31-0	Life Timer value
LTL	Shows the value of timer 0 at the time LTMR64H was last read. It will only update if LTMR64H is read.

34.3.5 Timer Load Value Register (LDVAL0 - LDVAL1)

34.3.5.1 Offset

Register	Offset
LDVAL0	100h
LDVAL1	110h

34.3.5.2 Function

These registers select the timeout period for the timer interrupts.

Access: User read/write

34.3.5.3 Diagram

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

34.3.5.4 Fields

Field	Function
31-0 TSV	Timer Start Value Sets the timer start value. The timer counts down until it reaches 0, then generates an interrupt and loads this register value again. Writing a new value to this register does not restart the timer; instead the value is 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.

34.3.6 Current Timer Value Register (CVAL0 - CVAL1)

34.3.6.1 Offset

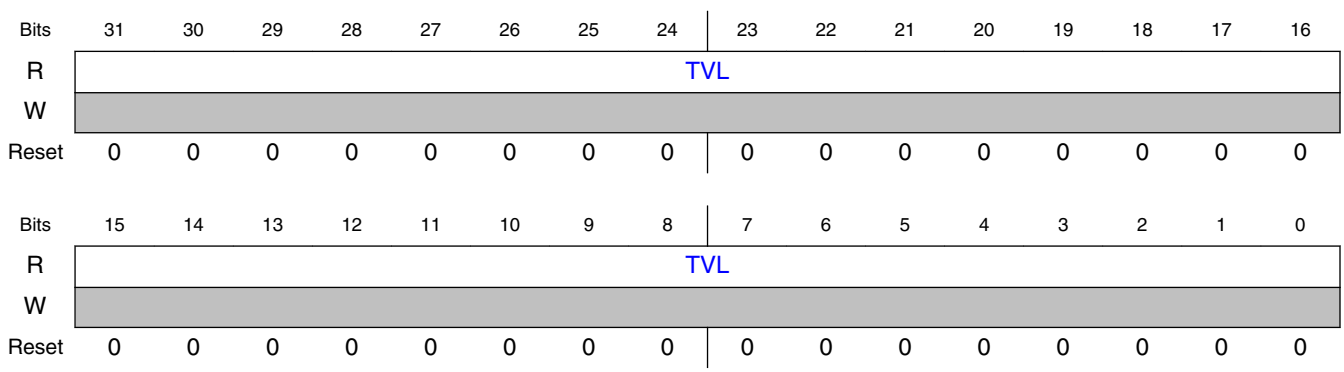
Register	Offset
CVAL0	104h
CVAL1	114h

34.3.6.2 Function

These registers indicate the current timer position.

Access: User read only

34.3.6.3 Diagram



34.3.6.4 Fields

Field	Function
31-0 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 because its value is unreliable. • The timer uses a downcounter. The timer values are frozen in the Debug mode if MCR[FRZ] is set.

34.3.7 Timer Control Register (TCTRL0 - TCTRL1)

34.3.7.1 Offset

Register	Offset
TCTRL0	108h
TCTRL1	118h

34.3.7.2 Function

These registers contain the control bits for each timer.

Access: User read/write

34.3.7.3 Diagram

Bits	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

Bits	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

34.3.7.4 Fields

Field	Function
31-3 —	Reserved
2 CHN	Chain Mode When activated, timer n-1 needs to expire before timer n can decrement by 1. Timer 0 cannot be chained. 0b - Timer is not chained. 1b - Timer is chained to a previous timer. For example, for channel 2, if this field is set, Timer 2 is chained to Timer 1.
1 TIE	Timer Interrupt Enable

Table continues on the next page...

Field	Function
	When an interrupt is pending, or if TFLGn[TIF] is set, enabling the interrupt causes an interrupt event. To avoid this, the associated TFLGn[TIF] must be cleared first. 0b - Interrupt requests from Timer n are disabled. 1b - Interrupt is requested whenever TIF is set.
0 TEN	Timer Enable Enables or disables the timer. 0b - Timer n is disabled. 1b - Timer n is enabled.

34.3.8 Timer Flag Register (TFLG0 - TFLG1)

34.3.8.1 Offset

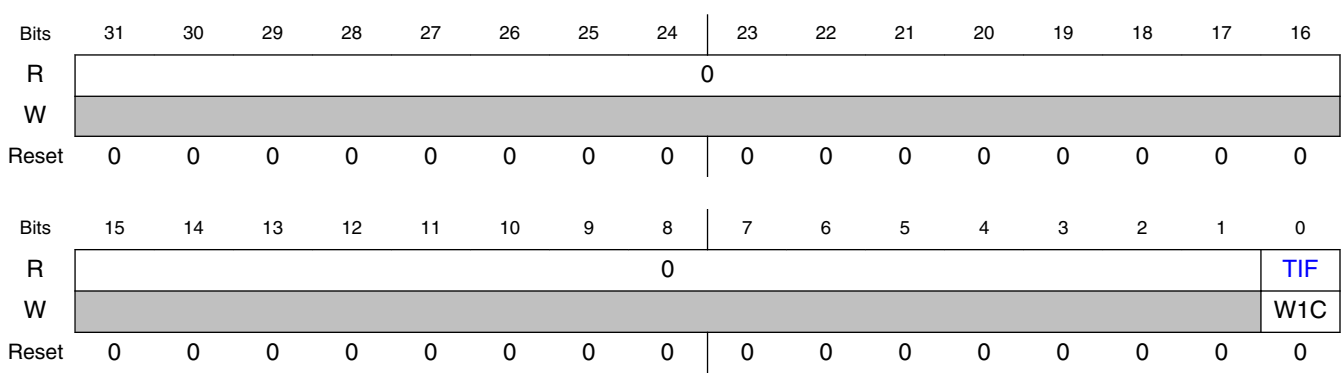
Register	Offset
TFLG0	10Ch
TFLG1	11Ch

34.3.8.2 Function

These registers hold the PIT interrupt flags.

Access: User read/write

34.3.8.3 Diagram



34.3.8.4 Fields

Field	Function
31-1 —	Reserved
0 TIF	Timer Interrupt Flag Sets to 1 at the end of the timer period. Writing 1 to this flag clears it and writing 0 has no effect. If enabled, or, when TCTRLn[TIE] = 1, TIF causes an interrupt request. 0b - Timeout has not yet occurred. 1b - Timeout has occurred.

34.4 Functional description

This section provides the functional description of the module.

34.4.1 General operation

This section provides detailed information on the internal operation of the module. Each timer can be used to generate trigger pulses and interrupts, and each interrupt is available on a separate interrupt line.

34.4.1.1 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 values again. Each time a timer reaches 0, it generates a trigger pulse and sets 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.

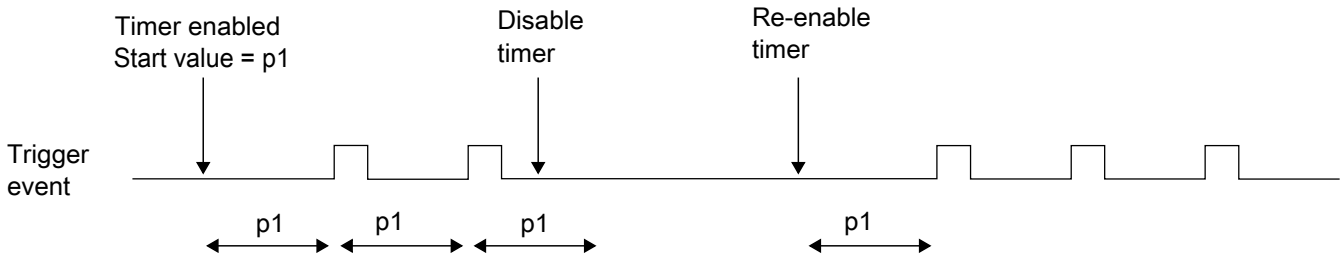


Figure 34-2. Stopping and starting a timer

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.

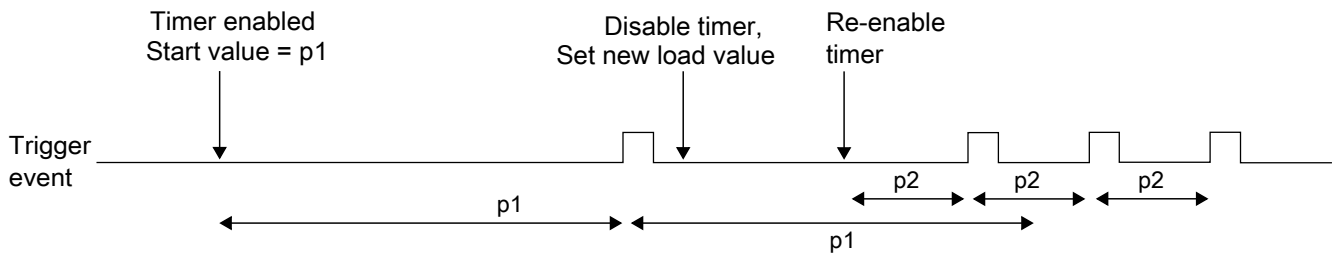


Figure 34-3. Modifying running timer period

It is also possible to change the counter period without restarting the timer, by writing LDVAL with the new load value. This value then loads after the next trigger event. See the following figure.

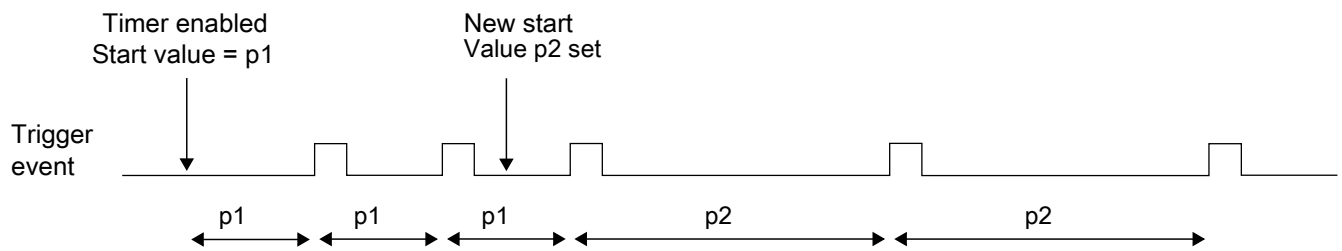


Figure 34-4. Dynamically setting a new load value

NOTE

- If the pause is initiated, when the timer is nearing 0 (CVALn =0x0), the pause command may not make it to the IP before the timer expires and generates a trigger.

- Pause will be ignored if (CVALn =0x0), but the trigger will remain asserted until the pause is removed. The user is recommended to remove Pause and then clear the interrupt TFLGn[TIF].
- If the timers are to be paused, sufficient time must be ensured for the IP to react.

34.4.1.2 Debug mode

In the Debug mode, the timers are 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.

34.4.2 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].

34.4.3 Chained timers

When a timer has chain mode enabled, it counts after the previous timer has expired. So if timer n-1 counts down to 0, counter n decrements 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.

34.5 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]. Also, 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
```

34.6 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 is 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
```

34.7 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, and 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 35
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.

35.1 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.

35.1.1 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

35.1.2 Modes of operation

The following table describes the operation of the LPTMR module in various modes.

Table 35-1. 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.

35.2 LPTMR signal descriptions

Table 35-2. LPTMR signal descriptions

Signal	I/O	Description
LPTMR0_ALT*n*	I	Pulse Counter Input pin

35.2.1 Detailed signal descriptions

Table 35-3. 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.

35.3 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	35.3.1/849
4004_0004	Low Power Timer Prescale Register (LPTMR0_PSR)	32	R/W	0000_0000h	35.3.2/850
4004_0008	Low Power Timer Compare Register (LPTMR0_CMR)	32	R/W	0000_0000h	35.3.3/852
4004_000C	Low Power Timer Counter Register (LPTMR0_CNR)	32	R/W	0000_0000h	35.3.4/852

35.3.1 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 chip-specific LPTMR information 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.

35.3.2 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.

35.3.3 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.

35.3.4 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

35.4 Functional description

35.4.1 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.

35.4.2 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.

35.4.3 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.

35.4.3.1 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.

35.4.3.2 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.

35.4.3.3 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.

35.4.3.4 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.

35.4.4 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.

35.4.5 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.

35.4.6 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.

35.4.7 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 36
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.

36.1 Introduction

36.1.1 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

36.1.2 Modes of operation

The RTC remains functional in all low power modes and can generate an interrupt to exit any low power mode.

36.1.3 RTC signal descriptions

Table 36-1. RTC signal descriptions

Signal	Description	I/O
RTC_CLKOUT	1 Hz square-wave output	O

36.1.3.1 RTC clock output

The clock to the seconds counter is available on the RTC_CLKOUT signal. It is a 1 Hz square wave output.

36.2 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	36.2.1/858
4003_D004	RTC Time Prescaler Register (RTC_TPR)	32	R/W	0000_0000h	36.2.2/859
4003_D008	RTC Time Alarm Register (RTC_TAR)	32	R/W	0000_0000h	36.2.3/859
4003_D00C	RTC Time Compensation Register (RTC_TCR)	32	R/W	0000_0000h	36.2.4/860
4003_D010	RTC Control Register (RTC_CR)	32	R/W	0000_0000h	36.2.5/861
4003_D014	RTC Status Register (RTC_SR)	32	R/W	0000_0001h	36.2.6/863
4003_D018	RTC Lock Register (RTC_LR)	32	R/W	0000_00FFh	36.2.7/864
4003_D01C	RTC Interrupt Enable Register (RTC_IER)	32	R/W	0000_0007h	36.2.8/865

36.2.1 RTC Time Seconds Register (RTC_TSR)

Address: 4003_D000h base + 0h offset = 4003_D000h

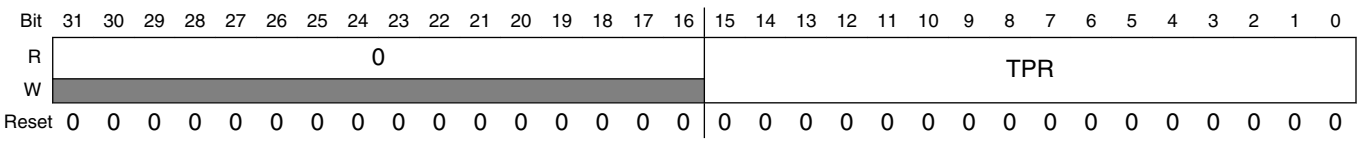
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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).

36.2.2 RTC Time Prescaler Register (RTC_TPR)

Address: 4003_D000h base + 4h offset = 4003_D004h

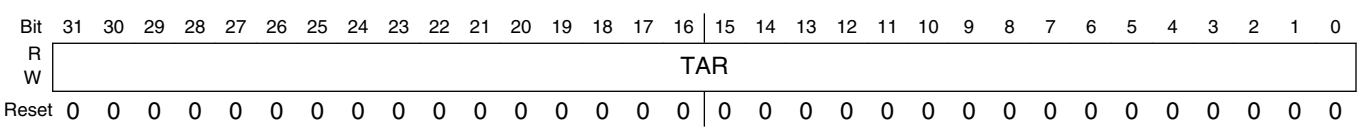


RTC_TPR field descriptions

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.

36.2.3 RTC Time Alarm Register (RTC_TAR)

Address: 4003_D000h base + 8h offset = 4003_D008h

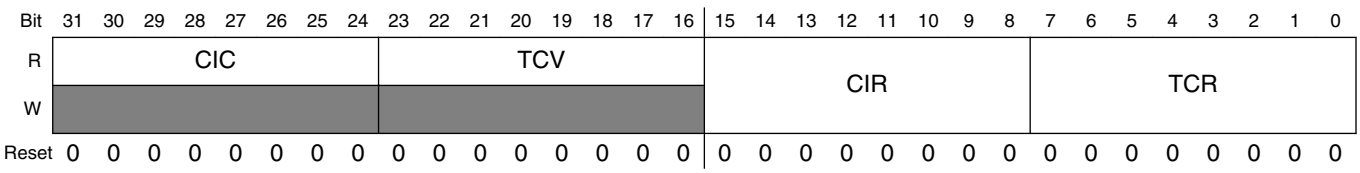


RTC_TAR field descriptions

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].

36.2.4 RTC Time Compensation Register (RTC_TCR)

Address: 4003_D000h base + Ch offset = 4003_D00Ch

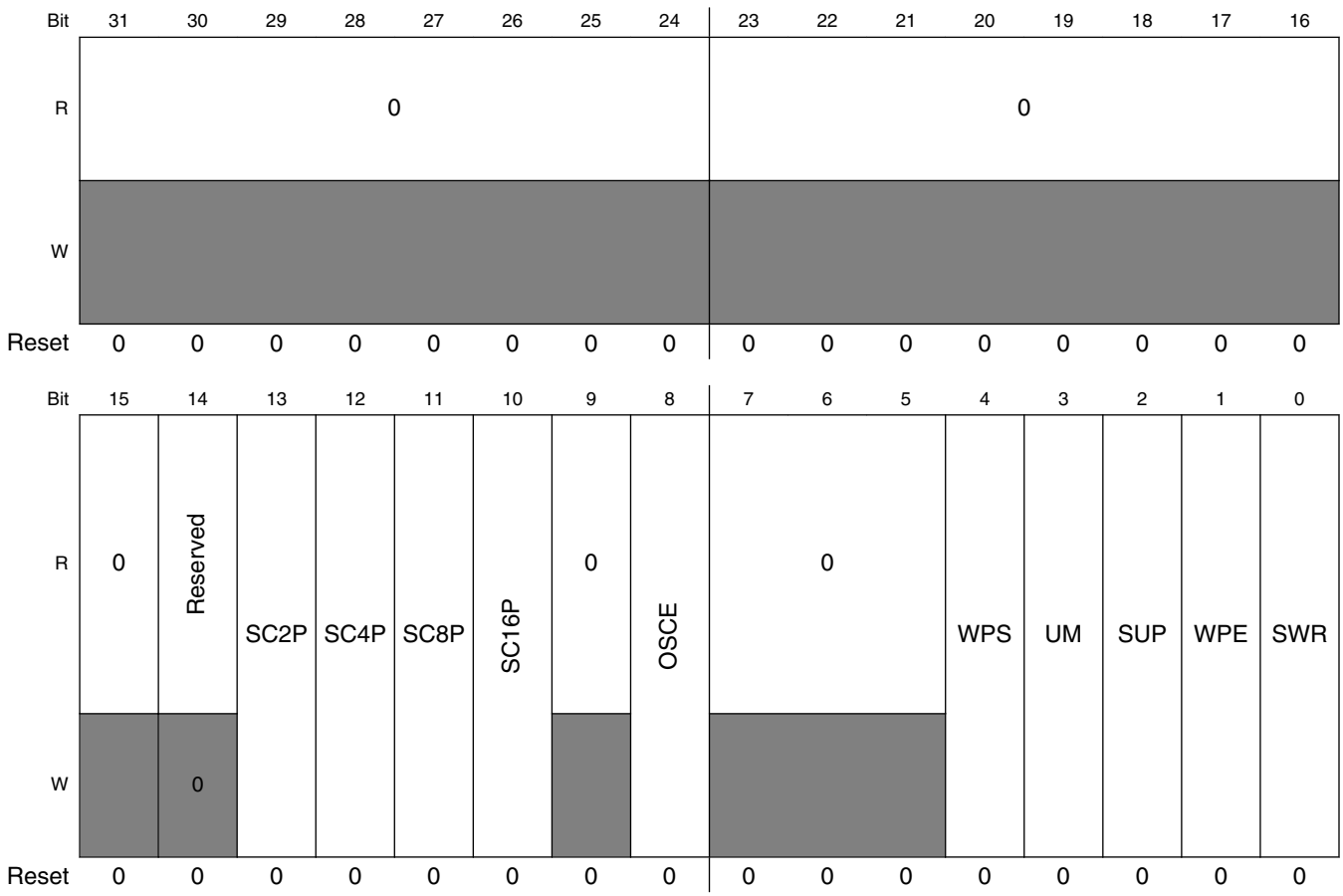


RTC_TCR field descriptions

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.

36.2.5 RTC Control Register (RTC_CR)

Address: 4003_D000h base + 10h offset = 4003_D010h



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

36.2.6 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

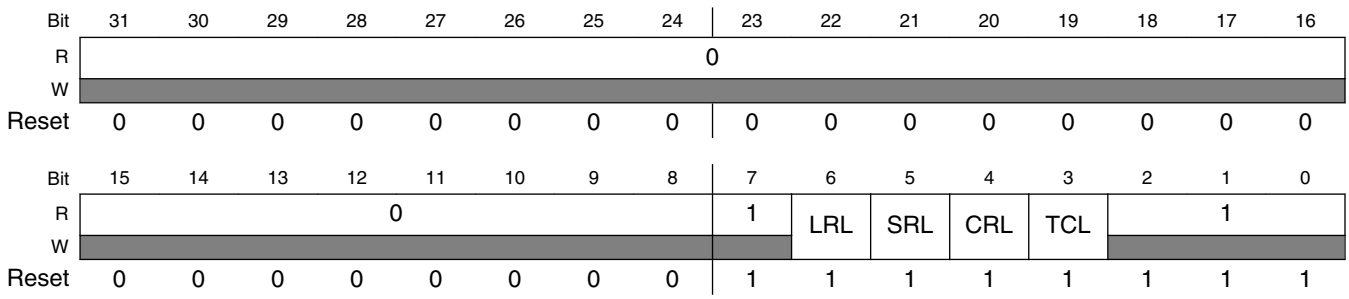
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											TCE	0	TAF	TOF	TIF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

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.

36.2.7 RTC Lock Register (RTC_LR)

Address: 4003_D000h base + 18h offset = 4003_D018h

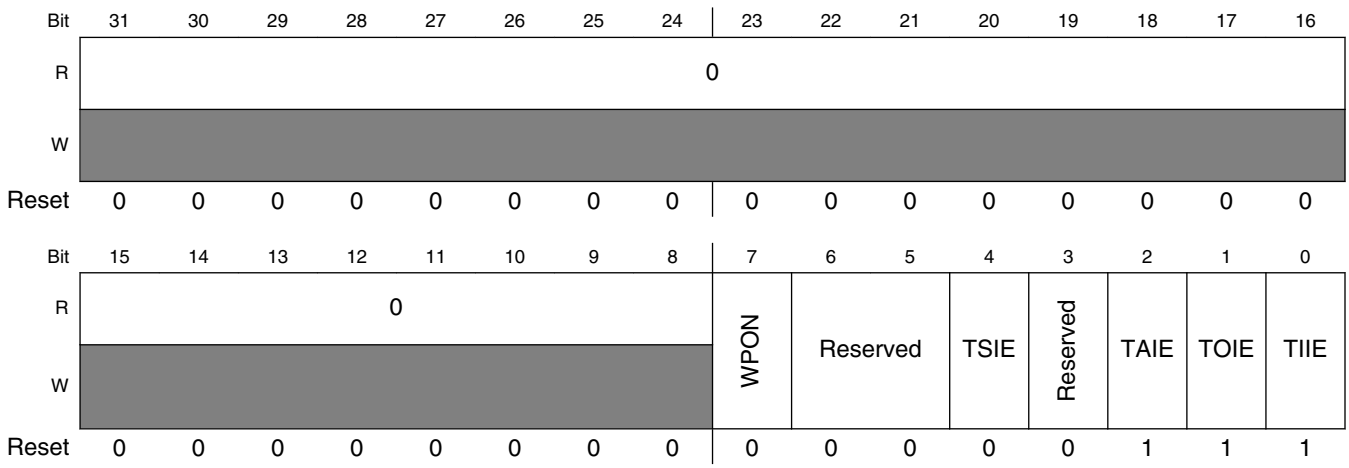


RTC_LR field descriptions

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.

36.2.8 RTC Interrupt Enable Register (RTC_IER)

Address: 4003_D000h base + 1Ch offset = 4003_D01Ch



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.

36.3 Functional description

36.3.1 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.

36.3.1.1 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.

36.3.1.2 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.

36.3.1.3 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.

36.3.2 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.

36.3.3 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.

36.3.4 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].

36.3.5 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].

36.3.6 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.

36.3.7 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 37
FlexCAN

37.1 Introduction

The FlexCAN module is a communication controller implementing the CAN protocol according to the ISO 11898-1 standard and CAN 2.0 B protocol specifications. A general block diagram is shown in the following figure, which describes the main subblocks implemented in the FlexCAN module, including one associated memory for storing message buffers, Receive Global Mask registers, Receive Individual Mask registers, Receive FIFO filters, and Receive FIFO ID filters. The functions of the submodules are described in subsequent sections.

NOTE

Rx FIFO cannot be used in FD mode.

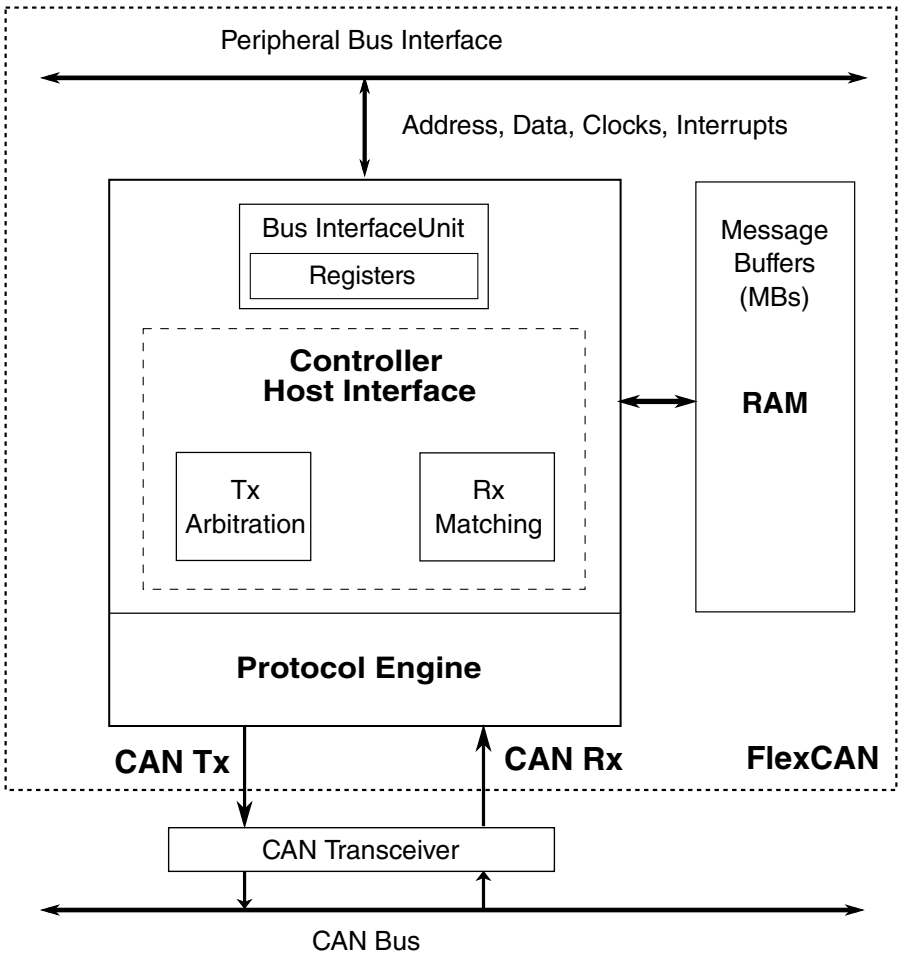


Figure 37-1. FlexCAN block diagram

37.1.1 Overview

The CAN protocol was primarily designed to be used as a vehicle serial data bus, meeting the specific requirements of this field:

- Real-time processing
- Reliable operation in the EMI environment of a vehicle
- Cost-effectiveness
- Required bandwidth

The FlexCAN module is a full implementation of the CAN protocol specification, the CAN with Flexible Data rate (CAN FD) protocol and the CAN 2.0 version B protocol, which supports both standard and extended message frames and long payloads up to 64 bytes transferred at faster rates up to 8 Mbps. The message buffers are stored in an embedded RAM dedicated to the FlexCAN module. See the chip configuration details for the actual number of message buffers configured in the chip.

The Protocol Engine (PE) submodule manages the serial communication on the CAN bus:

- Requesting RAM access for receiving and transmitting message frames
- Validating received messages
- Performing error handling
- Detecting CAN FD messages

The Controller Host Interface (CHI) sub-module handles message buffer selection for reception and transmission, taking care of arbitration and ID matching algorithms for both CAN FD and non-CAN FD message formats.

The Bus Interface Unit (BIU) sub-module controls the access to and from the internal interface bus, in order to establish connection to the CPU and to other blocks. Clocks, address and data buses, interrupt outputs, DMA and test signals are accessed through the BIU.

37.1.2 FlexCAN module features

The FlexCAN module includes these distinctive features:

- Full implementation of the CAN with Flexible Data Rate (CAN FD) protocol specification and CAN protocol specification, Version 2.0 B
  - Standard data frames
  - Extended data frames
  - Zero to sixty four bytes data length
  - Programmable bit rate (see the chip-specific FlexCAN information for the specific maximum rate configuration)
  - Content-related addressing
- Compliant with the ISO 11898-1 standard
- Flexible mailboxes configurable to store 0 to 8, 16, 32 or 64 bytes data length
- Each mailbox configurable as receive or transmit, all supporting standard and extended messages
- Individual Rx Mask registers per mailbox
- Full-featured Rx FIFO with storage capacity for up to six frames and automatic internal pointer handling with DMA support

- Transmission abort capability
- Flexible message buffers (MBs), totaling 32 message buffers of 8 bytes data length each, configurable as Rx or Tx
- Programmable clock source to the CAN Protocol Interface, either peripheral clock or oscillator clock
- RAM not used by reception or transmission structures can be used as general purpose RAM space
- Listen-Only mode capability
- Programmable Loop-Back mode supporting self-test operation
- Programmable transmission priority scheme: lowest ID, lowest buffer number, or highest priority
- Time stamp based on 16-bit free-running timer, with an optional external time tick
- Global network time, synchronized by a specific message
- Maskable interrupts
- Independence from the transmission medium (an external transceiver is assumed)
- Short latency time due to an arbitration scheme for high-priority messages
- Low power modes, with programmable wake up on bus activity
- Transceiver Delay Compensation feature when transmitting CAN FD messages at faster data rates
- Remote request frames may be handled automatically or by software
- CAN bit time settings and configuration bits can only be written in Freeze mode
- Tx mailbox status (Lowest priority buffer or empty buffer)
- Identifier Acceptance Filter Hit Indicator (IDHIT) register for received frames
- SYNCH bit available in Error in Status 1 register to inform that the module is synchronous with CAN bus
- CRC status for transmitted message
- Rx FIFO Global Mask register
- Selectable priority between mailboxes and Rx FIFO during matching process

- Powerful Rx FIFO ID filtering, capable of matching incoming IDs against either 128 extended, 256 standard, or 512 partial (8 bit) IDs, with up to 32 individual masking capability
- 100% backward compatibility with previous FlexCAN version

37.1.3 Modes of operation

The FlexCAN module has these functional modes:

- Normal mode (User or Supervisor):

  In Normal mode, the module operates receiving and/or transmitting message frames, errors are handled normally, and all CAN Protocol functions are enabled. User and Supervisor Modes differ in the access to some restricted control registers.

- Freeze mode:

  Freeze mode is enabled when the FRZ bit in MCR is asserted. If enabled, Freeze mode is entered when MCR[HALT] is set or when Debug mode is requested at chip level and MCR[FRZ_ACK] is asserted by the FlexCAN. In this mode, no transmission or reception of frames is done and synchronicity to the CAN bus is lost. See Freeze mode for more information.

- Loop-Back mode:

  The module enters this mode when the LPB field in the Control 1 Register is asserted. In this mode, FlexCAN performs an internal loop back that can be used for self-test operation. The bit stream output of the transmitter is internally fed back to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic '1'). FlexCAN behaves as it normally does when transmitting and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated.

- Listen-Only mode:

  The module enters this mode when the LOM field in the Control 1 Register is asserted. In this mode, transmission is disabled, all error counters are frozen, and the module operates in a CAN Error Passive mode. Only messages acknowledged by

another CAN station will be received. If FlexCAN detects a message that has not been acknowledged, it will flag a BIT0 error (without changing the REC), as if it was trying to acknowledge the message.

- CAN FD Active mode:

  In this mode, FlexCAN is capable of transmitting and receiving all messages formatted according to the CAN FD Protocol and CAN 2.0 Protocol 2.0 in a interleaved fashion. The CPU can set the FlexCAN into CAN FD Active mode by configuring the MCR[FDEN] bit field in Freeze Mode.

  It is important to know which features are available in CAN FD active mode. This table lists the differences between FD and classical modes.

Table 37-1. Differences between CAN classical and CAN FD

Feature	CAN classical	CAN FD
Rx FIFO DMA	Yes	No
Rx FIFO	Yes	No

For low-power operation, the FlexCAN module has:

- Module Disable mode:

  This low-power mode is entered when the MDIS bit in the MCR Register is asserted by the CPU and the LPM_ACK is asserted by the FlexCAN. When disabled, the module requests to disable the clocks to the CAN Protocol Engine and Controller Host Interface submodules. Exit from this mode is done by negating the MDIS bit in the MCR register. See Module Disable mode for more information.

- Doze mode:

  This low power mode is entered when the DOZE bit in MCR is asserted and Doze mode is requested at chip level and the LPM_ACK bit in the MCR Register is asserted by the FlexCAN. When in Doze mode, the module requests to disable the clocks to the CAN Protocol Engine and the CAN Controller-Host Interface sub-modules. Exit from this mode happens when the DOZE bit in MCR is negated, when the chip is removed from Doze mode, or when activity is detected on the CAN bus and the Self Wake Up mechanism is enabled. See Doze mode for more information.

- Stop mode:

  This low power mode is entered when Stop mode is requested at chip level and the LPM_ACK bit in the MCR Register is asserted by the FlexCAN. When in Stop Mode, the module puts itself in an inactive state and then informs the CPU that the

clocks can be shut down globally. Exit from this mode happens when the Stop mode request is removed, or when activity is detected on the CAN bus and the Self Wake Up mechanism is enabled. See Stop mode for more information.

37.2 FlexCAN signal descriptions

The FlexCAN module has two I/O signals connected to the external chip pins. These signals are summarized in the following table and described in more detail in the next subsections.

Table 37-2. FlexCAN signal descriptions

Signal	Description	I/O
CAN Rx	CAN Receive Pin	Input
CAN Tx	CAN Transmit Pin	Output

37.2.1 CAN Rx

This pin is the receive pin from the CAN bus transceiver. Dominant state is represented by logic level 0. Recessive state is represented by logic level 1.

37.2.2 CAN Tx

This pin is the transmit pin to the CAN bus transceiver. Dominant state is represented by logic level 0. Recessive state is represented by logic level 1.

37.3 Memory map/register definition

This section describes the registers and data structures in the FlexCAN module. The base address of the module depends on the particular memory map of the chip.

37.3.1 FlexCAN memory mapping

The memory map for the FlexCAN module is shown in the following table.

The address space occupied by FlexCAN has 128 bytes for registers starting at the module base address, followed by embedded RAM starting at address offset 0x0080.

Each individual register is identified by its complete name and the corresponding mnemonic. The access type can be Supervisor (S) or Unrestricted (U). Most of the registers can be configured to have either Supervisor or Unrestricted access by programming the SUPV field in the MCR register. These registers are identified as S/U in the Access column of Table 37-3.

NOTE

An invalid register access will result in a bus error. This includes reading a write-only register, writing a read-only register or accessing an invalid address.

Table 37-3. Register access and reset information

Register	Access type	Affected by hard reset	Affected by soft reset
Module Configuration Register (CAN_MCR)	S	Yes	Yes
Control 1 register (CAN_CTRL1)	S/U	Yes	No
Free Running Timer register (CAN_TIMER)	S/U	Yes	Yes
Rx Mailboxes Global Mask register (CAN_RXMGMASK)	S/U	No	No
Rx Buffer 14 Mask register (CAN_RX14MASK)	S/U	No	No
Rx Buffer 15 Mask register (CAN_RX15MASK)	S/U	No	No
Error Counter Register (CAN_ECR)	S/U	Yes	Yes
Error and Status 1 Register (CAN_ESR1)	S/U	Yes	Yes
Interrupt Masks 1 register (CAN_IMASK1)	S/U	Yes	Yes
Interrupt Flags 1 register (CAN_IFLAG1)	S/U	Yes	Yes
Control 2 Register (CAN_CTRL2)	S/U	Yes	No
Error and Status 2 Register (CAN_ESR2)	S/U	Yes	Yes
CRC Register (CAN_CRCR)	S/U	Yes	Yes
Rx FIFO Global Mask register (CAN_RXFGMASK)	S/U	No	No
Rx FIFO Information Register (CAN_RXFIR)	S/U	No	No
CAN Bit Timing Register (CAN_CBT)	S/U	Yes	No
Message buffers	S/U	No	No
Rx Individual Mask Registers	S/U	No	No
CAN FD Control register (CAN_FDCTRL)	S/U	Yes	No
CAN FD Bit Timing register (CAN_FDCBT)	S/U	Yes	No
CAN FD CRC register (CAN_FDCRC)	S/U	Yes	Yes

The FlexCAN module can store CAN messages for transmission and reception using mailboxes and Rx FIFO structures.

37.3.2 CAN register descriptions

The table below shows the FlexCAN memory map.

The address range from offset 0x80 to 0x27F allocates the thirty-two 128-bit Message Buffers (MBs).

The memory maps for the message buffers are in FlexCAN message buffer memory map.

37.3.2.1 CAN Memory map

CAN0 base address: 4002_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Module Configuration Register (MCR)	32	RW	D890_000Fh
4h	Control 1 register (CTRL1)	32	RW	0000_0000h
8h	Free Running Timer (TIMER)	32	RW	0000_0000h
10h	Rx Mailboxes Global Mask Register (RXMGMASK)	32	RW	See description.
14h	Rx 14 Mask register (RX14MASK)	32	RW	See description.
18h	Rx 15 Mask register (RX15MASK)	32	RW	See description.
1Ch	Error Counter (ECR)	32	RW	0000_0000h
20h	Error and Status 1 register (ESR1)	32	W1C	0000_0000h
28h	Interrupt Masks 1 register (IMASK1)	32	RW	0000_0000h
30h	Interrupt Flags 1 register (IFLAG1)	32	W1C	0000_0000h
34h	Control 2 register (CTRL2)	32	RW	00A0_0000h
38h	Error and Status 2 register (ESR2)	32	RO	0000_0000h
44h	CRC Register (CRCR)	32	RO	0000_0000h
48h	Rx FIFO Global Mask register (RXFGMASK)	32	RW	See description.
4Ch	Rx FIFO Information Register (RXFIR)	32	RO	See description.
50h	CAN Bit Timing Register (CBT)	32	RW	0000_0000h
880h - 8FCh	Rx Individual Mask Registers (RXIMR0 - RXIMR31)	32	RW	See description.
C00h	CAN FD Control Register (FDCTRL)	32	RW	8000_0100h
C04h	CAN FD Bit Timing Register (FDCBT)	32	RW	0000_0000h
C08h	CAN FD CRC Register (FDCRC)	32	RO	0000_0000h

37.3.2.2 Module Configuration Register (MCR)

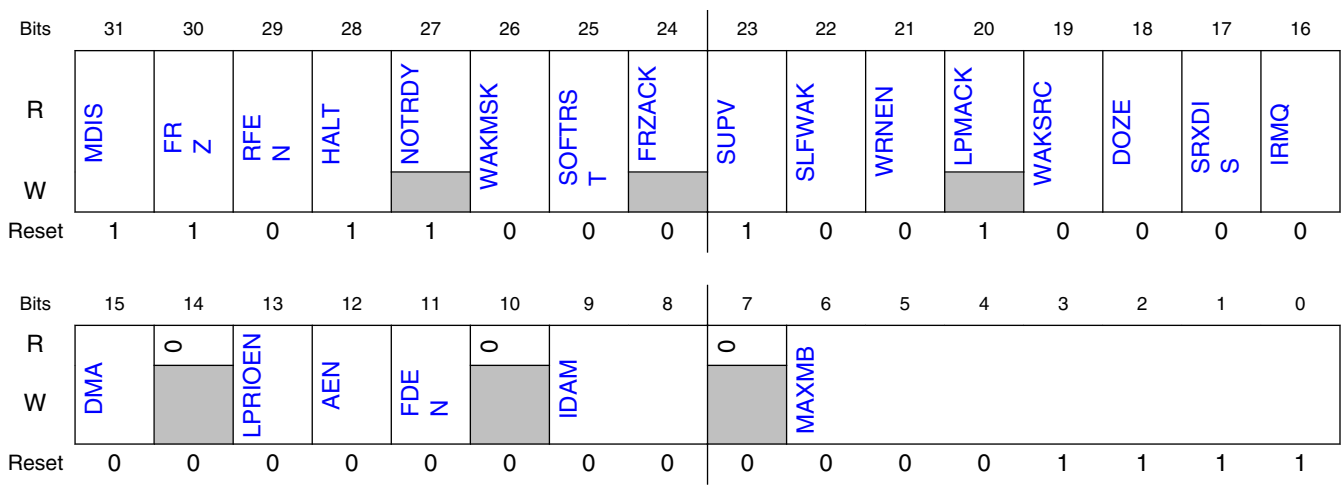
37.3.2.2.1 Offset

Register	Offset
MCR	0h

37.3.2.2.2 Function

This register defines global system configurations, such as the module operation modes and the maximum message buffer configuration.

37.3.2.2.3 Diagram



37.3.2.2.4 Fields

Field	Function
31 MDIS	Module Disable This bit controls whether FlexCAN is enabled or not. When disabled, FlexCAN disables the clocks to the CAN Protocol Engine and Controller Host Interface sub-modules. This bit is not affected by soft reset. 0b - Enable the FlexCAN module. 1b - Disable the FlexCAN module.
30 FRZ	Freeze Enable

Table continues on the next page...

Field	Function
	The FRZ bit specifies the FlexCAN behavior when CAN_MCR[HALT] is set or when Debug mode is requested at chip level. When FRZ is asserted, FlexCAN is enabled to enter Freeze mode. Negation of this bit field causes FlexCAN to exit from Freeze mode. 0b - Not enabled to enter Freeze mode. 1b - Enabled to enter Freeze mode.
29 RFEN	Rx FIFO Enable This bit controls whether the Rx FIFO feature is enabled or not. When RFEN is set, MBs 0 to 5 cannot be used for normal reception and transmission because the corresponding memory region (0x80-0xDC) is used by the FIFO engine as well as additional MBs (up to 32, depending on CAN_CTRL2[RFFN] setting) which are used as Rx FIFO ID Filter Table elements. RFEN also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in the table "Minimum Ratio Between Peripheral Clock Frequency and CAN Bit Rate" (see Arbitration and matching timing). This bit can be written in Freeze mode only because it is blocked by hardware in other modes. **NOTE:** This bit cannot be set when CAN FD operation is enabled (see FDEN bit). 0b - Rx FIFO not enabled. 1b - Rx FIFO enabled.
28 HALT	Halt FlexCAN Assertion of this bit puts the FlexCAN module into Freeze mode. The CPU should clear it after initializing the Message Buffers and the Control Registers CTRL1 and CTRL2. No reception or transmission is performed by FlexCAN before this bit is cleared. Freeze mode cannot be entered while FlexCAN is in a low power mode. 0b - No Freeze mode request. 1b - Enters Freeze mode if the FRZ bit is asserted.
27 NOTRDY	FlexCAN Not Ready This read-only bit indicates that FlexCAN is either in Disable mode, Doze mode, Stop mode or Freeze mode. It is negated once FlexCAN has exited these modes. This bit is not affected by soft reset. 0b - FlexCAN module is either in Normal mode, Listen-Only mode or Loop-Back mode. 1b - FlexCAN module is either in Disable mode, Doze mode, Stop mode or Freeze mode.
26 WAKMSK	Wake Up Interrupt Mask This bit enables the Wake Up Interrupt generation under Self Wake Up mechanism. 0b - Wake Up Interrupt is disabled. 1b - Wake Up Interrupt is enabled.
25 SOFTRST	Soft Reset When this bit is asserted, FlexCAN resets its internal state machines and some of the memory mapped registers. The SOFTRST bit can be asserted directly by the CPU when it writes to the MCR Register. Because soft reset is synchronous and has to follow a request/acknowledge procedure across clock domains, it may take some time to fully propagate its effect. The SOFTRST bit remains asserted while reset is pending, and is automatically negated when reset completes. Therefore, software can poll this bit to know when the soft reset has completed. Soft reset cannot be applied while clocks are shut down in a low power mode. The module should be first removed from low power mode, and then soft reset can be applied. This bit is not affected by soft reset. 0b - No reset request. 1b - Resets the registers affected by soft reset.
24 FRZACK	Freeze Mode Acknowledge This read-only bit indicates that FlexCAN is in Freeze mode and its prescaler is stopped. The Freeze mode request cannot be granted until current transmission or reception processes have finished. Therefore the software can poll the FRZACK bit to know when FlexCAN has actually entered Freeze mode. If Freeze Mode request is negated, then this bit is negated after the FlexCAN prescaler is running

Table continues on the next page...

Field	Function
	again. If Freeze mode is requested while FlexCAN is in a low power mode, then the FRZACK bit will be set only when the low-power mode is exited. See Section "Freeze Mode". This bit is not affected by soft reset. **NOTE:** FRZACK will be asserted within 178 CAN bits from the freeze mode request by the CPU, and negated within 2 CAN bits after the freeze mode request removal (see Protocol timing. 0b - FlexCAN not in Freeze mode, prescaler running. 1b - FlexCAN in Freeze mode, prescaler stopped.
23 SUPV	Supervisor Mode This bit configures the FlexCAN to be either in Supervisor or User mode. The registers affected by this bit are marked as S/U in the Access Type column of the module memory map. Reset value of this bit is 1, so the affected registers start with Supervisor access allowance only. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - FlexCAN is in User mode. Affected registers allow both Supervisor and Unrestricted accesses. 1b - FlexCAN is in Supervisor mode. Affected registers allow only Supervisor access. Unrestricted access behaves as though the access was done to an unimplemented register location.
22 SLFWAK	Self Wake Up This bit enables the Self Wake Up feature when FlexCAN is in a low-power mode other than Disable mode. When this feature is enabled, the FlexCAN module monitors the bus for wake up event, that is, a recessive-to-dominant transition. If a wake up event is detected during Doze mode, FlexCAN requests to resume its clocks and, if enabled to do so, generates a Wake Up interrupt to the CPU. If a wake up event is detected during Stop mode, then FlexCAN generates, if enabled to do so, a Wake Up interrupt to the CPU so that it can exit Stop mode globally and FlexCAN can request to resume the clocks. When FlexCAN is in a low-power mode other than Disable mode, this bit cannot be written as it is blocked by hardware. 0b - FlexCAN Self Wake Up feature is disabled. 1b - FlexCAN Self Wake Up feature is enabled.
21 WRNEN	Warning Interrupt Enable When asserted, this bit enables the generation of the TWRNINT and RWRNINT flags in the Error and Status Register 1 (ESR1). If WRNEN is negated, the TWRNINT and RWRNINT flags will always be zero, independent of the values of the error counters, and no warning interrupt will ever be generated. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0b - TWRNINT and RWRNINT bits are zero, independent of the values in the error counters. 1b - TWRNINT and RWRNINT bits are set when the respective error counter transitions from less than 96 to greater than or equal to 96.
20 LPMACK	Low-Power Mode Acknowledge This read-only bit indicates that FlexCAN is in a low-power mode (Disable mode, Doze mode, Stop mode). A low-power mode cannot be entered until all current transmission or reception processes have finished, so the CPU can poll the LPMACK bit to know when FlexCAN has actually entered low power mode. This bit is not affected by soft reset. **NOTE:** LPMACK will be asserted within 180 CAN bits from the low-power mode request by the CPU, and negated within 2 CAN bits after the low-power mode request removal (see Protocol timing). 0b - FlexCAN is not in a low-power mode. 1b - FlexCAN is in a low-power mode.
19 WAKSRC	Wake Up Source This bit defines whether the integrated low-pass filter is applied to protect the Rx CAN input from spurious wake up. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - FlexCAN uses the unfiltered Rx input to detect recessive to dominant edges on the CAN bus.

Table continues on the next page...

Field	Function
	1b - FlexCAN uses the filtered Rx input to detect recessive to dominant edges on the CAN bus.
18 DOZE	Doze Mode Enable This bit defines whether FlexCAN is allowed to enter low-power mode when Doze mode is requested at chip level . This bit is automatically reset when FlexCAN wakes up from Doze mode upon detecting activity on the CAN bus (self wake-up enabled). 0b - FlexCAN is not enabled to enter low-power mode when Doze mode is requested. 1b - FlexCAN is enabled to enter low-power mode when Doze mode is requested.
17 SRXDIS	Self Reception Disable This bit defines whether FlexCAN is allowed to receive frames transmitted by itself. If this bit is asserted, frames transmitted by the module will not be stored in any MB, regardless if the MB is programmed with an ID that matches the transmitted frame, and no interrupt flag or interrupt signal will be generated due to the frame reception. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Self reception enabled. 1b - Self reception disabled.
16 IRMQ	Individual Rx Masking And Queue Enable This bit indicates whether Rx matching process will be based either on individual masking and queue or on masking scheme with RXMGMASK, RX14MASK, RX15MASK and RXFGMASK. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0b - Individual Rx masking and queue feature are disabled. For backward compatibility with legacy applications, the reading of C/S word locks the MB even if it is EMPTY. 1b - Individual Rx masking and queue feature are enabled.
15 DMA	DMA Enable The DMA Enable bit controls whether the DMA feature is enabled or not. The DMA feature can only be used in Rx FIFO, consequently CAN_MCR[RFEN] must be asserted. When DMA and RFEN are set, CAN_IFLAG1[BUF5I] generates the DMA request and no RX FIFO interrupt is generated. This bit can be written in Freeze mode only as it is blocked by hardware in other modes. 0b - DMA feature for RX FIFO disabled. 1b - DMA feature for RX FIFO enabled.
14 —	Reserved
13 LPRIOEN	Local Priority Enable This bit is provided for backwards compatibility with legacy applications. It controls whether the local priority feature is enabled or not. It is used to expand the ID used during the arbitration process. With this expanded ID concept, the arbitration process is done based on the full 32-bit word, but the actual transmitted ID still has 11-bit for standard frames and 29-bit for extended frames. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Local Priority disabled. 1b - Local Priority enabled.
12 AEN	Abort Enable When asserted, this bit enables the Tx abort mechanism. This mechanism guarantees a safe procedure for aborting a pending transmission, so that no frame is sent in the CAN bus without notification. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. **NOTE:** When CAN_MCR[AEN] is asserted, only the abort mechanism (see Transmission abort mechanism) must be used for updating Mailboxes configured for transmission. **CAUTION:** Writing the Abort code into Rx Mailboxes can cause unpredictable results when CAN_MCR[AEN] is asserted. 0b - Abort disabled. 1b - Abort enabled.
11	CAN FD operation enable

Table continues on the next page...

Field	Function
FDEN	This bit enables the CAN with Flexible Data rate (CAN FD) operation. This bit can be written in Freeze mode only. **NOTE:** FlexCAN is able to transmit FD frame format according to ISO 11898-1. **NOTE:** The Rx FIFO Enable (RFEN) bit cannot be set if FDEN is asserted. 0b - CAN FD is disabled. FlexCAN is able to receive and transmit messages in CAN 2.0 format. 1b - CAN FD is enabled. FlexCAN is able to receive and transmit messages in both CAN FD and CAN 2.0 formats.
10 —	Reserved
9-8 IDAM	ID Acceptance Mode This 2-bit field identifies the format of the Rx FIFO ID Filter Table elements. Note that all elements of the table are configured at the same time by this field (they are all the same format). See Section "Rx FIFO Structure". This field can be written only in Freeze mode because it is blocked by hardware in other modes. 00b - Format A: One full ID (standard and extended) per ID Filter Table element. 01b - Format B: Two full standard IDs or two partial 14-bit (standard and extended) IDs per ID Filter Table element. 10b - Format C: Four partial 8-bit Standard IDs per ID Filter Table element. 11b - Format D: All frames rejected.
7 —	Reserved
6-0 MAXMB	Number Of The Last Message Buffer This 7-bit field defines the number of the last Message Buffers that will take part in the matching and arbitration processes. The reset value (0x0F) is equivalent to a 16 MB configuration. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Number of the last MB = MAXMB **NOTE:** MAXMB must be programmed with a value smaller than or equal to the number of available Message Buffers, as described in FlexCAN Memory Partition for CAN FD. Additionally, the definition of MAXMB value must take into account the region of MBs occupied by Rx FIFO and its ID filters table space defined by CAN_CTRL2[RFFN]. MAXMB also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table "Minimum Ratio Between Peripheral Clock Frequency and CAN Bit Rate" (see Arbitration and matching timing).

37.3.2.3 Control 1 register (CTRL1)

37.3.2.3.1 Offset

Register	Offset
CTRL1	4h

37.3.2.3.2 Function

This register is defined for specific FlexCAN control features related to the CAN bus, such as bit-rate, programmable sampling point within an Rx bit, Loop Back mode, Listen-Only mode, Bus Off recovery behavior and interrupt enabling (Bus-Off, Error, Warning). It also determines the Division Factor for the clock prescaler.

The CAN bit timing variables (PRESDIV, PROPSEG, PSEG1, PSEG2 and RJW) can also be configured in the CBT register, which extends the range of all these variables. If CAN_CBT[BTF] is set, PRESDIV, PROPSEG, PSEG1, PSEG2 and RJW fields of CAN_CTRL1 become read only.

NOTE

When the CAN FD feature is enabled, do not use the PRESDIV, RJW, PSEG1, PSEG2, and PROPSEG fields of the CTRL1 register for CAN bit timing. Instead use the CBT register's EPRESDIV, ERJW, EPSEG1, EPSEG2, and EPROPSEG fields.

The contents of this register are not affected by soft reset.

NOTE

The CAN bit variables in CTRL1 and in CBT are stored in the same register.

37.3.2.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PRESDIV								RJW		PSEG1			PSEG2		
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	BOFFMSK	ERRMSK	CLKSRC	LPB	TWRNMSK	RWRNMSK	Reserved	Reserved	SMP	BOFFREC	TSYN	LBUF	LOM	PROPSEG		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

37.3.2.3.4 Fields

Field	Function
31-24	Prescaler Division Factor

Table continues on the next page...

Field	Function
PRESDIV	This 8-bit field defines the ratio between the PE clock frequency and the Serial Clock (Sclock) frequency. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency. The Maximum value of this field is 0xFF, that gives a minimum Sclock frequency equal to the PE clock frequency divided by 256. See Protocol timing. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (PRESDIV + 1)
23-22 RJW	Resync Jump Width This 2-bit field defines the maximum number of time quanta that a bit time can be changed by one re-synchronization. One time quantum is equal to the Sclock period. The valid programmable values are 0–3. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = RJW + 1.
21-19 PSEG1	Phase Segment 1 This 3-bit field defines the length of Phase Segment 1 in the bit time. The valid programmable values are 0–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 1 = (PSEG1 + 1) × Time-Quanta.
18-16 PSEG2	Phase Segment 2 This 3-bit field defines the length of Phase Segment 2 in the bit time. The valid programmable values are 1–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 2 = (PSEG2 + 1) × Time-Quanta.
15 BOFFMSK	Bus Off Interrupt Mask This bit provides a mask for the Bus Off Interrupt CAN_ESR1[BOFFINT]. 0b - Bus Off interrupt disabled. 1b - Bus Off interrupt enabled.
14 ERRMSK	Error Interrupt Mask This bit provides a mask for the Error Interrupt CAN_ESR1[ERRINT]. 0b - Error interrupt disabled. 1b - Error interrupt enabled.
13 CLKSRC	CAN Engine Clock Source This bit selects the clock source to the CAN Protocol Engine (PE) to be either the peripheral clock or the oscillator clock. The selected clock is the one fed to the prescaler to generate the Serial Clock (Sclock). In order to guarantee reliable operation, this bit can be written only in Disable mode because it is blocked by hardware in other modes. See Protocol timing. **NOTE:** The user must ensure the protocol engine clock tolerance according to the CAN Protocol standard (ISO 11898-1). **NOTE:** See the clock distribution chapter (module clocks table) to identify the proper clock source. 0b - The CAN engine clock source is the oscillator clock. Under this condition, the oscillator clock frequency must be lower than the bus clock. 1b - The CAN engine clock source is the peripheral clock.
12 LPB	Loop Back Mode This bit configures FlexCAN to operate in Loop-Back mode. In this mode, FlexCAN performs an internal loop back that can be used for self test operation. The bit stream output of the transmitter is fed back internally to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic 1). FlexCAN behaves as it normally does when transmitting, and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field, generating an internal acknowledge bit to ensure proper reception of its own message. Both transmit and receive interrupts are generated. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.

Table continues on the next page...

Field	Function
	NOTE: In this mode, CAN_MCR[SRXDIS] cannot be asserted because this will impede the self reception of a transmitted message. **NOTE:** CAN_FDCTRL[TDCEN] must be disabled when LPB is asserted. 0b - Loop Back disabled. 1b - Loop Back enabled.
11 TWRNMSK	Tx Warning Interrupt Mask This bit provides a mask for the Tx Warning Interrupt associated with the TWRNINT flag in the Error and Status Register 1 (ESR1). This bit is read as zero when CAN_MCR[WRNEN] is negated. This bit can be written only if CAN_MCR[WRNEN] is asserted. 0b - Tx Warning Interrupt disabled. 1b - Tx Warning Interrupt enabled.
10 RWRNMSK	Rx Warning Interrupt Mask This bit provides a mask for the Rx Warning Interrupt associated with the RWRNINT flag in the Error and Status Register 1 (ESR1). This bit is read as zero when MCR[WRNEN] bit is negated. This bit can be written only if CAN_MCR[WRNEN] bit is asserted. 0b - Rx Warning Interrupt disabled. 1b - Rx Warning Interrupt enabled.
9 —	Reserved
8 —	Reserved
7 SMP	CAN Bit Sampling This bit defines the sampling mode of CAN bits at the Rx input. It can be written in Freeze mode only because it is blocked by hardware in other modes. **NOTE:** For proper operation, to assert SMP it is necessary to guarantee a minimum value of 2 TQs in CAN_CTRL1[PSEG1] (or CAN_CBT[EPSEG1]).This bit cannot be asserted when CAN FD is enabled (CAN_MCR[FDEN] = 1). 0b - Just one sample is used to determine the bit value. 1b - Three samples are used to determine the value of the received bit: the regular one (sample point) and 2 preceding samples; a majority rule is used.
6 BOFFREC	Bus Off Recovery This bit defines how FlexCAN recovers from Bus Off state. If this bit is negated, automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. If the bit is asserted, automatic recovering from Bus Off is disabled and the module remains in Bus Off state until the bit is negated by the user. If the negation occurs before 128 sequences of 11 recessive bits are detected on the CAN bus, then Bus Off recovery happens as if the BOFFREC bit had never been asserted. If the negation occurs after 128 sequences of 11 recessive bits occurred, then FlexCAN will re-synchronize to the bus by waiting for 11 recessive bits before joining the bus. After negation, the BOFFREC bit can be re-asserted again during Bus Off, but it will be effective only the next time the module enters Bus Off. If BOFFREC was negated when the module entered Bus Off, asserting it during Bus Off will not be effective for the current Bus Off recovery. **NOTE:** Refer to Bus off in the CAN Protocol standard (ISO 11898-1) for details. 0b - Automatic recovering from Bus Off state enabled. 1b - Automatic recovering from Bus Off state disabled.
5 TSYN	Timer Sync This bit enables a mechanism that resets the free-running timer each time a message is received in Message Buffer 0. This feature provides means to synchronize multiple FlexCAN stations with a special “SYNC” message, that is, global network time. If CAN_MCR[RFEN] is set (Rx FIFO enabled), the first available Mailbox, according to CAN_CTRL2[RFFN] setting, is used for timer synchronization instead of MB0. This bit can be written in Freeze mode only because it is blocked by hardware in other modes.

Table continues on the next page...

Field	Function
	0b - Timer Sync feature disabled 1b - Timer Sync feature enabled
4 LBUF	Lowest Buffer Transmitted First This bit defines the ordering mechanism for Message Buffer transmission. When asserted, CAN_MCR[LPRIOEN] does not affect the priority arbitration. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0b - Buffer with highest priority is transmitted first. 1b - Lowest number buffer is transmitted first.
3 LOM	Listen-Only Mode This bit configures FlexCAN to operate in Listen-Only mode. In this mode, transmission is disabled, all error counters described in the ECR register are frozen and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FlexCAN detects a message that has not been acknowledged, it will flag a BIT0 error without changing the receive error counter (CAN_ECR[RXERRCNT]), as if it was trying to acknowledge the message. Listen-Only mode is acknowledged by the state of CAN_ESR1[FLTCONF] field indicating Passive Error. There can be some delay between the Listen-Only mode request and acknowledge. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0b - Listen-Only mode is deactivated. 1b - FlexCAN module operates in Listen-Only mode.
2-0 PROPSEG	Propagation Segment This 3-bit field defines the length of the Propagation Segment in the bit time. The valid programmable values are 0–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation Segment Time = (PROPSEG + 1) × Time-Quanta. Time-Quantum = one Sclock period.

37.3.2.4 Free Running Timer (TIMER)

37.3.2.4.1 Offset

Register	Offset
TIMER	8h

37.3.2.4.2 Function

This register represents a 16-bit free running counter that can be read and written by the CPU. The timer starts from 0x0 after Reset, counts linearly to 0xFFFF, and wraps around.

When CAN_CTRL2[TIMER_SRC] is asserted, the timer is continuously incremented by an external time tick. The time tick must be synchronous to the Peripheral Clock, with a minimum pulse width of one clock cycle.

When CAN_CTRL2[TIMER_SRC] is negated, the timer is incremented by the CAN bit clock, which defines the baud rate on the CAN bus. During a message transmission/ reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate. The timer is not incremented during Disable, Doze, Stop and Freeze modes.

The timer value is captured when the second bit of the identifier field of any frame is on the CAN bus. This captured value is written into the Time Stamp entry in a message buffer after a successful reception or transmission of a message.

If CTRL1[TSYN] is asserted, the Timer is reset whenever a message is received in the first available Mailbox, according to CAN_CTRL2[RFFN] setting.

The CPU can write to this register anytime. However, if the write occurs at the same time that the Timer is being reset by a reception in the first Mailbox, then the write value is discarded.

Reading this register affects the Mailbox Unlocking procedure, see Section "Mailbox Lock Mechanism".

37.3.2.4.3 Diagram

Bits	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

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

37.3.2.4.4 Fields

Field	Function
31-16 —	Reserved
15-0 TIMER	Timer Value Contains the free-running counter value.

37.3.2.5 Rx Mailboxes Global Mask Register (RXMGMASK)

37.3.2.5.1 Offset

Register	Offset
RXMGMASK	10h

37.3.2.5.2 Function

This register is located in RAM.

RXMGMASK is provided for legacy application support.

- When CAN_MCR[IRMQ] is negated, RXMGMASK is always in effect (the bits in the MG field will mask the Mailbox filter bits).
- When CAN_MCR[IRMQ] is asserted, RXMGMASK has no effect (the bits in the MG field will not mask the Mailbox filter bits).

RXMGMASK is used to mask the filter fields of all Rx MBs, excluding MBs 14-15, which have individual mask registers.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

37.3.2.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MG															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

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

37.3.2.5.4 Fields

Field	Function
31-0 MG	Rx Mailboxes Global Mask Bits

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>These bits mask the Mailbox filter bits. Note that the alignment with the ID word of the Mailbox is not perfect as the two most significant MG bits affect the fields RTR and IDE, which are located in the Control and Status word of the Mailbox. The following table shows in detail which MG bits mask each Mailbox filter field.
<table>
<tr><th rowspan="2">CAN_SMB[RTR]</th><th rowspan="2">CAN_CTRL2[RRS]</th><th rowspan="2">CAN_CTRL2[EACEN]</th><th colspan="4">Mailbox filter fields</th></tr>
<tr><th>MB[RTR]</th><th>MB[IDE]</th><th>MB[ID]</th><th>Reserved</th></tr>
<tr><td>0</td><td>-</td><td>0</td><td>note</td><td>note</td><td>MG[28:0]</td><td>MG[31:29]</td></tr>
<tr><td>0</td><td>-</td><td>1</td><td>MG[31]</td><td>MG[30]</td><td>MG[28:0]</td><td>MG[29]</td></tr>
<tr><td>1</td><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>MG[31:0]</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>-</td><td>-</td><td>MG[28:0]</td><td>MG[31:29]</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>MG[31]</td><td>MG[30]</td><td>MG[28:0]</td><td>MG[29]</td></tr>
</table>
1. RTR bit of the Incoming Frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).
2. If the CAN_CTRL2[EACEN] bit is negated, the RTR bit of Mailbox is never compared with the RTR bit of the incoming frame.
3. If CAN_CTRL2[EACEN] is negated, the IDE bit of Mailbox is always compared with the IDE bit of the incoming frame.

0b - The corresponding bit in the filter is "don't care."
1b - The corresponding bit in the filter is checked.</td></tr>
</table>

1. RTR bit of the Incoming Frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).
2. If the CAN_CTRL2[EACEN] bit is negated, the RTR bit of Mailbox is never compared with the RTR bit of the incoming frame.
3. If CAN_CTRL2[EACEN] is negated, the IDE bit of Mailbox is always compared with the IDE bit of the incoming frame.

37.3.2.6 Rx 14 Mask register (RX14MASK)

37.3.2.6.1 Offset

Register	Offset
RX14MASK	14h

37.3.2.6.2 Function

This register is located in RAM.

RX14MASK is provided for legacy application support. When CAN_MCR[IRMQ] is asserted, RX14MASK has no effect.

RX14MASK is used to mask the filter fields of Message Buffer 14.

This register can only be programmed while the module is in Freeze mode as it is blocked by hardware in other modes.

37.3.2.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX14M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX14M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

37.3.2.6.4 Fields

Field	Function
31-0 RX14M	Rx Buffer 14 Mask Bits Each mask bit masks the corresponding Mailbox 14 filter field in the same way that RXMGMASK masks other Mailboxes' filters. See the description of the RXMGMASK register. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

37.3.2.7 Rx 15 Mask register (RX15MASK)

37.3.2.7.1 Offset

Register	Offset
RX15MASK	18h

37.3.2.7.2 Function

This register is located in RAM.

RX15MASK is provided for legacy application support. When CAN_MCR[IRMQ] is asserted, RX15MASK has no effect.

RX15MASK is used to mask the filter fields of Message Buffer 15.

This register can be programmed only while the module is in Freeze mode because it is blocked by hardware in other modes.

37.3.2.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX15M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX15M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

37.3.2.7.4 Fields

Field	Function
31-0 RX15M	Rx Buffer 15 Mask Bits Each mask bit masks the corresponding Mailbox 15 filter field in the same way that RXMGMASK masks other Mailboxes' filters. See the description of the CAN_RXMGMASK register. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

37.3.2.8 Error Counter (ECR)

37.3.2.8.1 Offset

Register	Offset
ECR	1Ch

37.3.2.8.2 Function

This register has four 8-bit fields reflecting the value of the FlexCAN error counters:

- Transmit Error Counter (TXERRCNT field)

- Receive Error Counter (RXERRCNT field)
- Transmit Error Counter for errors detected in the Data Phase of CAN FD messages with the BRS bit set (TXERRCNT_FAST field)
- Receive Error Counter for errors detected in the Data Phase of CAN FD messages with the BRS bit set (RXERRCNT_FAST field)

The TXERRCNT and RXERRCNT counters take into account all errors in both CAN FD and non-FD message formats. TXERRCNT_FAST and RXERRCNT_FAST are dedicated to count only the errors occurred in the Data Phase of CAN FD frames with the BRS bit set.

The Fault Confinement State (FLTCONF field in Error and Status Register 1 - CAN_ESR1) is updated based on TXERRCNT and RXERRCNT counters only. TXERRCNT and RXERRCNT counters can be written in Freeze mode only. TXERRCNT_FAST and RXERRCNT_FAST counters are read-only except in Freeze mode where the CPU can write value zero. The rules for increasing and decreasing these counters are described in the CAN protocol and are completely implemented in the FlexCAN module.

The following are the basic rules for FlexCAN bus state transitions:

- If the value of TXERRCNT or RXERRCNT increases to be greater than or equal to 128, the FLTCONF field in the Error and Status Register is updated to reflect "Error Passive" state.
- If the FlexCAN state is "Error Passive", and either TXERRCNT or RXERRCNT decrements to a value less than or equal to 127 while the other already satisfies this condition, the FLTCONF field in the Error and Status Register is updated to reflect "Error Active" state.
- If the value of TXERRCNT increases to be greater than 255, the FLTCONF field in the Error and Status Register is updated to reflect "Bus Off" state, and an interrupt may be issued. The value of TXERRCNT is then reset to zero.
- If FlexCAN is in "Bus Off' state, then TXERRCNT is cascaded together with another internal counter to count the 128th occurrences of 11 consecutive recessive bits on the bus. Hence, TXERRCNT is reset to zero and counts in a manner where the internal counter counts 11 such bits and then wraps around while incrementing the TXERRCNT. When TXERRCNT reaches the value of 128, the FLTCONF field in the Error and Status Register is updated to be "Error Active" and both error counters are reset to zero. At any instance of dominant bit following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero without affecting the TXERRCNT value. The TXERRCNT_FAST counter is frozen during busoff.
- If during system start-up, only one node is operating, then its TXERRCNT increases in each message it is trying to transmit, as a result of acknowledge errors (indicated by the ACKERR bit in the Error and Status Register). After the transition to "Error

Passive" state, the TXERRCNT does not increment anymore by acknowledge errors. Therefore the device never goes to the "Bus Off" state.

- If the RXERRCNT increases to a value greater than 127, it is not incremented further, even if more errors are detected while being a receiver. At the next successful message reception, the counter is set to a value between 119 and 127 to resume to "Error Active" state.
- TXERRCNT_FAST and RXERRCNT_FAST error counters values increment and decrement based on errors detected only in the Data Phase of CAN FD frames with the BRS bit set, following the same increment and decrement rules as TXERRCNT and RXERRCNT counters. These counters do not wrap around and get stuck at their maximum value (255). They stop counting and keep their values frozen while FlexCAN is in "Bus Off" state. They are reset when FlexCAN leaves "Bus Off" state and restart counting once FlexCAN resumes to "Error Active" state.

NOTE

Refer to Fault confinement in the CAN Protocol standard (ISO 11898-1) for details.

37.3.2.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXERRCNT_FAST								TXERRCNT_FAST							
W	0								0							
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	RXERRCNT								TXERRCNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

37.3.2.8.4 Fields

Field	Function
31-24 RXERRCNT_FAST	Receive Error Counter for fast bits Receive Error Counter for errors detected in the Data Phase of received CAN FD messages with the BRS bit set. The RXERRCNT_FAST counter is read-only except in Freeze mode, where the CPU can write a 8-bit zero value only.
23-16 TXERRCNT_FAST	Transmit Error Counter for fast bits Transmit Error Counter for errors detected in the Data Phase of transmitted CAN FD messages with the BRS bit set. The TXERRCNT_FAST counter is read-only except in Freeze mode, where the CPU can write a 8-bit zero value only.

Table continues on the next page...

Field	Function
15-8 RXERRCNT	Receive Error Counter Receive Error Counter for all errors detected in received messages. The RXERRCNT counter is read-only except in Freeze mode, where it can be written by the CPU.
7-0 TXERRCNT	Transmit Error Counter Transmit Error Counter for all errors detected in transmitted messages. The TXERRCNT counter is read-only except in Freeze mode, where it can be written by the CPU.

37.3.2.9 Error and Status 1 register (ESR1)

37.3.2.9.1 Offset

Register	Offset
ESR1	20h

37.3.2.9.2 Function

This register reports various error conditions detected in the reception and transmission of a CAN frame, some general status of the device and it is the source of some interrupts to the CPU.

The reported error conditions are BIT1ERR, BIT0ERR, ACKKERR, CRCERR, FRMERR and STFERR, for errors detected in CAN frames of any format, and BIT1ERR_FAST, BIT0ERR_FAST, CRCERR_FAST, FRMERR_FAST and STFERR_FAST for errors detected in the Data Phase of CAN FD frames with the BRS bit set only.

An error detected in a single CAN frame may be reported by one or more error flags. Also, error reporting is cumulative in case more error events happen in the next frames while the CPU does not attempt to read this register.

TXWRN, RXWRN, IDLE, TX, FLTCONF, RX and SYNCH are status bits.

BOFFINT, BOFFDONEINT, ERRINT, ERRINT_FAST, WAKINT, TWRNINT, and RWRNINT are interrupt bits. It is recommended the CPU to use the following procedure when servicing interrupt requests generated by these bits:

- Read this register to capture all error condition and status bits. This action clear the respective bits that were set since the last read access.
- Write 1 to clear the interrupt bit that has triggered the interrupt request.
- Write 1 to clear the ERR_OVR bit if it is set.

Starting from all error flags cleared, a first error event sets either the ERRINT or the ERRINT_FAST (provided the corresponding mask bit is asserted). If other error events in subsequent frames happen before the CPU to serve the interrupt request, the ERR_OVR bit is set to indicate that errors from different frames had accumulated.

SYNCH	IDLE	TX	RX	FlexCAN State
0	0	0	0	Not synchronized to CAN bus
1	1	x	x	Idle
1	0	1	0	Transmitting
1	0	0	1	Receiving

NOTE

Refer to Fault confinement in the CAN Protocol standard (ISO 11898-1) for details.

37.3.2.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BIT1ERR_FAST	BIT0ERR_FAST	0	CRCERR_FAST	FRMERR_FAST	STFERR_FAST	0				ERROVR	ERRINT_FAST	BOFFDONEINT	SYNCH	TWRNINT	RWRNINT
W											W1C	W1C	W1C		W1C	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	BIT1ER	BIT0ER	ACKERR	CRCER	FRMER	STFER	TXWRN	RXWRN	IDLE	TX	FLTCONF		RX	BOFFINT	ERRINT	WAKINT
W														W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

37.3.2.9.4 Fields

Field	Function
31 BIT1ERR_FAST	Bit1 Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates when an inconsistency occurs between the transmitted and the received bit in the Data Phase of CAN FD frames with the BRS bit set.

Table continues on the next page...

Field	Function
	0b - No such occurrence. 1b - At least one bit sent as recessive is received as dominant.
30 BIT0ERR_FAST	Bit0 Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates when an inconsistency occurs between the transmitted and the received bit in the Data Phase of CAN FD frames with the BRS bit set. 0b - No such occurrence. 1b - At least one bit sent as dominant is received as recessive.
29 —	Reserved
28 CRCERR_FAST	Cyclic Redundancy Check Error in the CRC field of CAN FD frames with the BRS bit set This bit indicates that a CRC Error has been detected by the receiver node in the CRC field of CAN FD frames with the BRS bit set, that is, the calculated CRC is different from the received. 0b - No such occurrence. 1b - A CRC error occurred since last read of this register.
27 FRMERR_FAST	Form Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates that a Form Error has been detected by the receiver node in the Data Phase of CAN FD frames with the BRS bit set, that is, a fixed-form bit field contains at least one illegal bit. 0b - No such occurrence. 1b - A Form Error occurred since last read of this register.
26 STFERR_FAST	Stuffing Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates that a Stuffing Error has been detected in the Data Phase of CAN FD frames with the BRS bit set. 0b - No such occurrence. 1b - A Stuffing Error occurred since last read of this register.
25-22 —	Reserved
21 ERROVR	Error Overrun bit This bit indicates that an error condition occurred when any error flag is already set. This bit is cleared by writing it to 1. 0b - Overrun has not occurred. 1b - Overrun has occurred.
20 ERRINT_FAST	Error Interrupt for errors detected in the Data Phase of CAN FD frames with the BRS bit set This bit indicates that at least one of the Error Bits detected in the Data Phase of CAN FD frames with the BRS bit set (BIT1ERR_FAST, BIT0ERR_FAST, CRCERR_FAST, FRMERR_FAST or STFERR_FAST) is set. If the corresponding mask bit CAN_CTRL2[ERRMSK_FAST] is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0b - No such occurrence. 1b - Indicates setting of any Error Bit detected in the Data Phase of CAN FD frames with the BRS bit set.
19 BOFFDONEINT	Bus Off Done Interrupt This bit is set when the Tx Error Counter (TXERRCNT) has finished counting 128 occurrences of 11 consecutive recessive bits on the CAN bus and is ready to leave Bus Off. If the corresponding mask bit in the Control 2 Register (BOFFDONEMSK) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0b - No such occurrence. 1b - FlexCAN module has completed Bus Off process.
18 SYNCH	CAN Synchronization Status

Table continues on the next page...

Field	Function
	This read-only flag indicates whether the FlexCAN is synchronized to the CAN bus and able to participate in the communication process. It is set and cleared by the FlexCAN. See the table in the overall CAN_ESR1 register description. 0b - FlexCAN is not synchronized to the CAN bus. 1b - FlexCAN is synchronized to the CAN bus.
17 TWRNINT	Tx Warning Interrupt Flag If the WRNEN bit in CAN_MCR is asserted, the TWRNINT bit is set when the TXWRN flag transitions from 0 to 1, meaning that the Tx error counter reached 96. If the corresponding mask bit in the Control 1 Register (CAN_CTRL1[TWRNMSK]) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When WRNEN is negated, this flag is masked. CPU must clear this flag before disabling the bit. Otherwise it will be set when the WRNEN is set again. Writing 0 has no effect. This flag is not generated during Bus Off state. This bit is not updated during Freeze mode. 0b - No such occurrence. 1b - The Tx error counter transitioned from less than 96 to greater than or equal to 96.
16 RWRNINT	Rx Warning Interrupt Flag If the WRNEN bit in CAN_MCR is asserted, the RWRNINT bit is set when the RXWRN flag transitions from 0 to 1, meaning that the Rx error counters reached 96. If the corresponding mask bit in the Control 1 Register (CAN_CTRL1[RWRNMSK]) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When WRNEN is negated, this flag is masked. CPU must clear this flag before disabling the bit. Otherwise it will be set when the WRNEN is set again. Writing 0 has no effect. This bit is not updated during Freeze mode. 0b - No such occurrence. 1b - The Rx error counter transitioned from less than 96 to greater than or equal to 96.
15 BIT1ERR	Bit1 Error This bit indicates when an inconsistency occurs between the transmitted and the received bit in a non-CAN FD message or in the arbitration or data phase of a CAN FD message. **NOTE:** This bit is not set by a transmitter in case of arbitration field or ACK slot, or in case of a node sending a passive error flag that detects dominant bits. 0b - No such occurrence. 1b - At least one bit sent as recessive is received as dominant.
14 BIT0ERR	Bit0 Error This bit indicates when an inconsistency occurs between the transmitted and the received bit in a non-CAN FD message or in the arbitration or data phase of a CAN FD message. 0b - No such occurrence. 1b - At least one bit sent as dominant is received as recessive.
13 ACKERR	Acknowledge Error This bit indicates that an Acknowledge Error has been detected by the transmitter node, that is, a dominant bit has not been detected during the ACK SLOT. 0b - No such occurrence. 1b - An ACK error occurred since last read of this register.
12 CRCERR	Cyclic Redundancy Check Error This bit indicates that a CRC Error has been detected by the receiver node either in a non-FD message or in the arbitration or data phase of a frame in CAN FD format, that is, the calculated CRC is different from the received. 0b - No such occurrence. 1b - A CRC error occurred since last read of this register.
11	Form Error

Table continues on the next page...

Field	Function
FRMERR	This bit indicates that a Form Error has been detected in a non-FD message or else in an FD message's arbitration or data phase by the receiver node, that is, a fixed-form bit field contains at least one illegal bit. 0b - No such occurrence. 1b - A Form Error occurred since last read of this register.
10 STFERR	Stuffing Error This bit indicates that a Stuffing Error has been detected in a non-FD message or else in an FD message's arbitration or data phase by the receiver node. 0b - No such occurrence. 1b - A Stuffing Error occurred since last read of this register.
9 TXWRN	TX Error Warning This bit indicates when repetitive errors are occurring during message transmission and is affected by the value of TXERRCNT in CAN_ECR register only. This bit is not updated during Freeze mode. 0b - No such occurrence. 1b - TXERRCNT is greater than or equal to 96.
8 RXWRN	Rx Error Warning This bit indicates when repetitive errors are occurring during message reception and is affected by the value of RXERRCNT in CAN_ECR register only. This bit is not updated during Freeze mode. 0b - No such occurrence. 1b - RXERRCNT is greater than or equal to 96.
7 IDLE	IDLE This bit indicates when CAN bus is in IDLE state. See the table in the overall CAN_ESR1 register description. 0b - No such occurrence. 1b - CAN bus is now IDLE.
6 TX	FlexCAN In Transmission This bit indicates if FlexCAN is transmitting a message. See the table in the overall CAN_ESR1 register description. 0b - FlexCAN is not transmitting a message. 1b - FlexCAN is transmitting a message.
5-4 FLTCONF	Fault Confinement State This 2-bit field indicates the Confinement State of the FlexCAN module. If the LOM bit in the Control Register 1 is asserted, after some delay that depends on the CAN bit timing the FLTCONF field will indicate “Error Passive”. The very same delay affects the way how FLTCONF reflects an update to CAN_ECR register by the CPU. It may be necessary up to one CAN bit time to get them coherent again. This bit field is affected by soft reset, but if the LOM bit is asserted, its reset value lasts just one CAN bit. After this time, FLTCONF reports "Error Passive". 00b - Error Active 01b - Error Passive 1xb - Bus Off
3 RX	FlexCAN In Reception This bit indicates if FlexCAN is receiving a message. See the table in the overall CAN_ESR1 register description. 0b - FlexCAN is not receiving a message. 1b - FlexCAN is receiving a message.
2	Bus Off Interrupt

Table continues on the next page...

Field	Function
BOFFINT	This bit is set when FlexCAN enters 'Bus Off' state. If the corresponding mask bit in the Control Register 1 (CAN_CTRL1[BOFFMSK]) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0b - No such occurrence. 1b - FlexCAN module entered Bus Off state.
1 ERRINT	Error Interrupt This bit indicates that at least one of the Error Bits (BIT1ERR, BIT0ERR, ACKERR. CRCERR, FRMERR or STFERR) is set. If the corresponding mask bit CAN_CTRL1[ERRMSK] is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0b - No such occurrence. 1b - Indicates setting of any Error Bit in the Error and Status Register.
0 WAKINT	Wake-Up Interrupt This field applies when FlexCAN is in low-power mode under Self Wake Up mechanism: • Doze mode • Stop mode When a recessive-to-dominant transition is detected on the CAN bus and if the CAN_MCR[WAKMSK] bit is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When CAN_MCR[SLFWAK] is negated, this flag is masked. The CPU must clear this flag before disabling the bit. Otherwise it will be set when the SLFWAK is set again. Writing 0 has no effect. 0b - No such occurrence. 1b - Indicates a recessive to dominant transition was received on the CAN bus.

37.3.2.10 Interrupt Masks 1 register (IMASK1)

37.3.2.10.1 Offset

Register	Offset
IMASK1	28h

37.3.2.10.2 Function

This register allows any number of a range of the 32 Message Buffer Interrupts to be enabled or disabled for MB31 to MB0. It contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception, that is, when the corresponding CAN_IFLAG1 bit is set.

37.3.2.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUF31TO0M															
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	BUF31TO0M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

37.3.2.10.4 Fields

Field	Function
31-0 BUF31TO0M	Buffer MB i Mask Each bit enables or disables the corresponding FlexCAN Message Buffer Interrupt for MB31 to MB0. **NOTE:** Setting or clearing a bit in the CAN_IMASK1 Register can assert or negate an interrupt request, if the corresponding IFLAG1 bit is set. 0b - The corresponding buffer Interrupt is disabled. 1b - The corresponding buffer Interrupt is enabled.

37.3.2.11 Interrupt Flags 1 register (IFLAG1)

37.3.2.11.1 Offset

Register	Offset
IFLAG1	30h

37.3.2.11.2 Function

This register defines the flags for the 32 Message Buffer interrupts for MB31 to MB0. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding CAN_IFLAG1 bit. If the corresponding CAN_IMASK1 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect. There is an exception when DMA for Rx FIFO is enabled, as described below.

The BUF7I to BUF5I flags are also used to represent FIFO interrupts when the Rx FIFO is enabled. When the bit CAN_MCR[RFEN] is set and the bit CAN_MCR[DMA] is negated, the function of the 8 least significant interrupt flags changes: BUF7I, BUF6I and BUF5I indicate operating conditions of the FIFO, BUF0I is used to empty FIFO, and BUF4I to BUF1I bits are reserved.

Before enabling the CAN_MCR[RFEN], the CPU must service the IFLAG bits asserted in the Rx FIFO region; see Section "Rx FIFO". Otherwise, these IFLAG bits will mistakenly show the related MBs now belonging to FIFO as having contents to be serviced. When the CAN_MCR[RFEN] bit is negated, the FIFO flags must be cleared. The same care must be taken when an CAN_CTRL2[RFFN] value is selected extending Rx FIFO filters beyond MB7. For example, when RFFN is 0x8, the MB0-23 range is occupied by Rx FIFO filters and related IFLAG bits must be cleared.

When both the CAN_MCR[RFEN] and CAN_MCR[DMA] bits are asserted (DMA feature for Rx FIFO enabled), the function of the 8 least significant interrupt flags (BUF7I - BUF0I) are changed to support the DMA operation. BUF7I and BUF6I are not used, as well as, BUF4I to BUF1I. BUF5I indicates operating condition of FIFO, and BUF0I is used to empty FIFO. Moreover, BUF5I does not generate a CPU interrupt, but generates a DMA request. IMASK1 bits in Rx FIFO region are not considered when bit CAN_MCR[DMA] is enabled. In addition the CPU must not clear the flag BUF5I when DMA is enabled. Before enabling the bit CAN_MCR[DMA], the CPU must service the IFLAGs asserted in the Rx FIFO region. When the bit CAN_MCR[DMA] is negated, the FIFO must be empty.FIFO must be disabled when FDEN bit in CAN_MCR register is enabled.

Before updating CAN_MCR[MAXMB] field, CPU must service the CAN_IFLAG1 bits whose MB value is greater than the CAN_MCR[MAXMB] to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.

37.3.2.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF31TO8I															
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	BUF31TO8I								BUF7I	BUF6I	BUF5I	BUF4TO1I				BUF0I
W	W1C								W1C	W1C	W1C	W1C				W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

37.3.2.11.4 Fields

Field	Function
31-8 BUF31TO8I	Buffer MBi Interrupt Each bit flags the corresponding FlexCAN Message Buffer interrupt for MB31 to MB8. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.
7 BUF7I	Buffer MB7 Interrupt Or "Rx FIFO Overflow" When the RFEN bit in the CAN_MCR register is cleared (Rx FIFO disabled), this bit flags the interrupt for MB7. **NOTE:** This flag is cleared by the FlexCAN whenever the bit CAN_MCR[RFEN] is changed by CPU writes. The BUF7I flag represents "Rx FIFO Overflow" when CAN_MCR[RFEN] is set. In this case, the flag indicates that a message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox. 0b - No occurrence of MB7 completing transmission/reception when CAN_MCR[RFEN]=0, or of Rx FIFO overflow when CAN_MCR[RFEN]=1 1b - MB7 completed transmission/reception when CAN_MCR[RFEN]=0, or Rx FIFO overflow when CAN_MCR[RFEN]=1
6 BUF6I	Buffer MB6 Interrupt Or "Rx FIFO Warning" When the RFEN bit in the CAN_MCR register is cleared (Rx FIFO disabled), this bit flags the interrupt for MB6. **NOTE:** This flag is cleared by the FlexCAN whenever the bit CAN_MCR[RFEN] is changed by CPU writes.

Table continues on the next page...

Field	Function
	The BUF6I flag represents "Rx FIFO Warning" when CAN_MCR[RFEN] is set. In this case, the flag indicates when the number of unread messages within the Rx FIFO is increased to 5 from 4 due to the reception of a new one, meaning that the Rx FIFO is almost full. Note that if the flag is cleared while the number of unread messages is greater than 4, it does not assert again until the number of unread messages within the Rx FIFO is decreased to be equal to or less than 4. 0b - No occurrence of MB6 completing transmission/reception when CAN_MCR[RFEN]=0, or of Rx FIFO almost full when CAN_MCR[RFEN]=1 1b - MB6 completed transmission/reception when CAN_MCR[RFEN]=0, or Rx FIFO almost full when CAN_MCR[RFEN]=1
5 BUF5I	Buffer MB5 Interrupt Or "Frames available in Rx FIFO" When the RFEN bit in the MCR is cleared (Rx FIFO disabled), this bit flags the interrupt for MB5. **NOTE:** This flag is cleared by the FlexCAN whenever the bit CAN_MCR[RFEN] is changed by CPU writes. When CAN_MCR[RFEN] is set (Rx FIFO enabled), the BUF5I flag represents "Frames available in Rx FIFO" and indicates that at least one frame is available to be read from the Rx FIFO. When the CAN_MCR[DMA] bit is enabled, this flag generates a DMA request and the CPU must not clear this bit by writing 1 in BUF5I. 0b - No occurrence of MB5 completing transmission/reception when CAN_MCR[RFEN]=0, or of frame(s) available in the FIFO, when CAN_MCR[RFEN]=1 1b - MB5 completed transmission/reception when CAN_MCR[RFEN]=0, or frame(s) available in the Rx FIFO when CAN_MCR[RFEN]=1. It generates a DMA request in case of CAN_MCR[RFEN] and CAN_MCR[DMA] are enabled.
4-1 BUF4TO1I	Buffer MB i Interrupt Or "reserved" When the RFEN bit in the CAN_MCR register is cleared (Rx FIFO disabled), these bits flag the interrupts for MB4 to MB1. **NOTE:** These flags are cleared by the FlexCAN whenever the bit CAN_MCR[RFEN] is changed by CPU writes. The BUF4TO1I flags are reserved when CAN_MCR[RFEN] is set. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception when CAN_MCR[RFEN]=0. 1b - The corresponding buffer has successfully completed transmission or reception when CAN_MCR[RFEN]=0.
0 BUF0I	Buffer MB0 Interrupt Or Clear FIFO bit When the RFEN bit in MCR is cleared (Rx FIFO disabled), this bit flags the interrupt for MB0. If the Rx FIFO is enabled, this bit is used to trigger the clear FIFO operation. This operation empties FIFO contents. Before performing this operation the CPU must service all FIFO related IFLAGs. When the bit CAN_MCR[DMA] is enabled this operation also clears the BUF5I flag and consequently abort the DMA request. The clear FIFO operation occurs when the CPU writes 1 in BUF0I. It is only allowed in Freeze Mode and is blocked by hardware in other conditions. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception when CAN_MCR[RFEN]=0. 1b - The corresponding buffer has successfully completed transmission or reception when CAN_MCR[RFEN]=0.

37.3.2.12 Control 2 register (CTRL2)

37.3.2.12.1 Offset

Register	Offset
CTRL2	34h

37.3.2.12.2 Function

This register complements Control1 Register providing control bits for memory write access in Freeze Mode, for extending FIFO filter quantity, and for adjust the operation of internal FlexCAN processes like matching and arbitration.

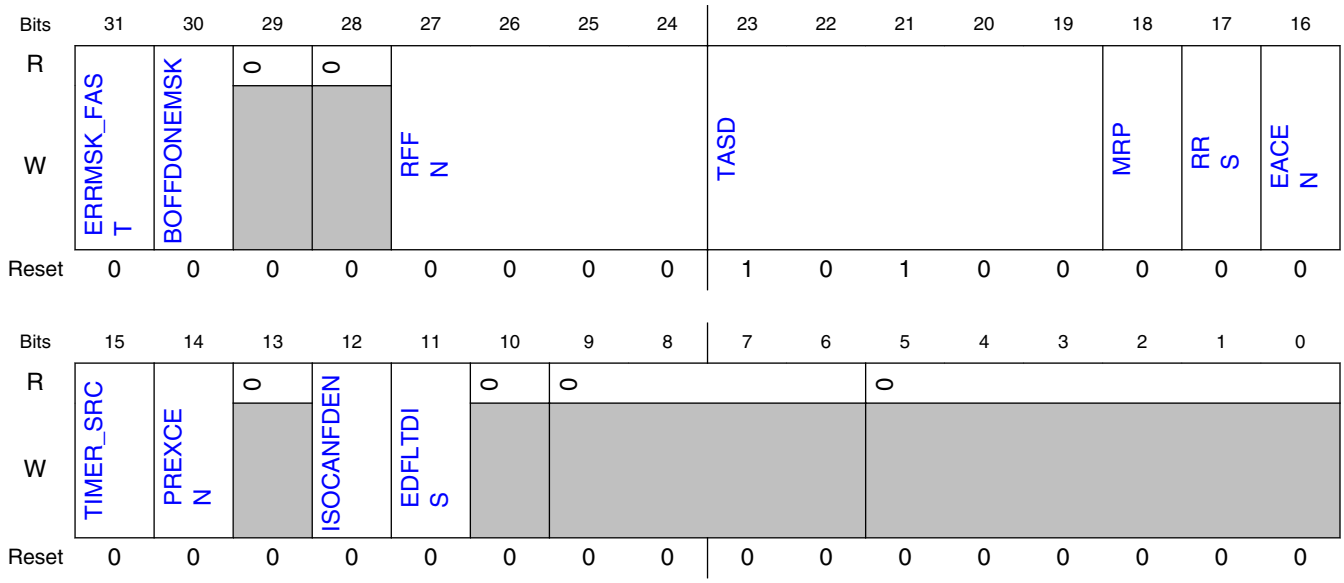
The contents of this register are not affected by soft reset.

This table shows how the Rx FIFO filter structure is determined by the value of CAN_CTRL2[RFFN]. See the CAN_CTRL2[RFFN] field description for more information.

Table 37-4. Rx FIFO filter: possible structures

RFFN[3:0]	Number of Rx FIFO filter elements	Message Buffers occupied by Rx FIFO and ID Filter Table	Remaining Available Mailboxes	Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks	Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask
0x0	8	MB 0-7	MB 8-31	Elements 0-7	none
0x1	16	MB 0-9	MB 10-31	Elements 0-9	Elements 10-15
0x2	24	MB 0-11	MB 12-31	Elements 0-11	Elements 12-23
0x3	32	MB 0-13	MB 14-31	Elements 0-13	Elements 14-31
0x4	40	MB 0-15	MB 16-31	Elements 0-15	Elements 16-39
0x5	48	MB 0-17	MB 18-31	Elements 0-17	Elements 18-47
0x6	56	MB 0-19	MB 20-31	Elements 0-19	Elements 20-55
0x7	64	MB 0-21	MB 22-31	Elements 0-21	Elements 22-63
0x8	72	MB 0-23	MB 24-31	Elements 0-23	Elements 24-71
0x9	80	MB 0-25	MB 26-31	Elements 0-25	Elements 26-79
0xA	88	MB 0-27	MB 28-31	Elements 0-27	Elements 28-87
0xB	96	MB 0-29	MB 30-31	Elements 0-29	Elements 30-95
0xC	104	MB 0-31	none	Elements 0-31	Elements 32-103

37.3.2.12.3 Diagram



37.3.2.12.4 Fields

Field	Function
31 ERRMSK_FAST	Error Interrupt Mask for errors detected in the Data Phase of fast CAN FD frames This bit provides a mask for the ERRINT_FAST Interrupt in CAN_ESR1 register. 0b - ERRINT_FAST Error interrupt disabled. 1b - ERRINT_FAST Error interrupt enabled.
30 BOFFDONEMSK	Bus Off Done Interrupt Mask This bit provides a mask for the Bus Off Done Interrupt in CAN_ESR1 register. 0b - Bus Off Done interrupt disabled. 1b - Bus Off Done interrupt enabled.
29 —	Reserved
28 —	Reserved
27-24 RFFN	Number Of Rx FIFO Filters This 4-bit field defines the number of Rx FIFO filters, as shown in Table 37-4. The maximum selectable number of filters is determined by the chip. This field can only be written in Freeze mode as it is blocked by hardware in other modes. This field must not be programmed with values that make the number of Message Buffers occupied by Rx FIFO and ID Filter exceed the number of Mailboxes present, defined by CAN_MCR[MAXMB]. **NOTE:** Each group of eight filters occupies a memory space equivalent to two Message Buffers which means that the more filters are implemented the less Mailboxes will be available. Considering that the Rx FIFO occupies the memory space originally reserved for MB0-5, RFFN should be programmed with a value corresponding to a number of filters not greater than the number of available memory words which can be calculated as follows: (SETUP_MB - 6) × 4

Table continues on the next page...

Field	Function
	where SETUP_MB is the least between the parameter NUMBER_OF_MB and CAN_MCR[MAXMB]. The number of remaining Mailboxes available will be: (SETUP_MB - 8) - (RFFN × 2) If the Number of Rx FIFO Filters programmed through RFFN exceeds the SETUP_MB value (memory space available) the exceeding ones will not be functional. **NOTE:** • The number of the last remaining available mailboxes is defined by the least value between the NUMBER_OF_MB minus 1 and the CAN_MCR[MAXMB] field. • If Rx Individual Mask Registers are not enabled then all Rx FIFO filters are affected by the Rx FIFO Global Mask.
23-19 TASD	Tx Arbitration Start Delay This 5-bit field indicates how many CAN bits the Tx arbitration process start point can be delayed from the first bit of CRC field on CAN bus. See Tx Arbitration start delay for more details. This field can be written only in Freeze mode because it is blocked by hardware in other modes.
18 MRP	Mailboxes Reception Priority If this bit is set the matching process starts from the Mailboxes and if no match occurs the matching continues on the Rx FIFO. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Matching starts from Rx FIFO and continues on Mailboxes. 1b - Matching starts from Mailboxes and continues on Rx FIFO.
17 RRS	Remote Request Storing If this bit is asserted Remote Request Frame is submitted to a matching process and stored in the corresponding Message Buffer in the same fashion of a Data Frame. No automatic Remote Response Frame will be generated. If this bit is negated the Remote Request Frame is submitted to a matching process and an automatic Remote Response Frame is generated if a Message Buffer with CODE=0b1010 is found with the same ID. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Remote Response Frame is generated. 1b - Remote Request Frame is stored.
16 EACEN	Entire Frame Arbitration Field Comparison Enable For Rx Mailboxes This bit controls the comparison of IDE and RTR bits within Rx Mailboxes filters with their corresponding bits in the incoming frame by the matching process. This bit does not affect matching for Rx FIFO. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Rx Mailbox filter's IDE bit is always compared and RTR is never compared despite mask bits. 1b - Enables the comparison of both Rx Mailbox filter's IDE and RTR bit with their corresponding bits within the incoming frame. Mask bits do apply.
15 TIMER_SRC	Timer Source Selects the time tick source used for incrementing the Free Running Timer counter. This bit can be written in Freeze mode only. 0b - The Free Running Timer is clocked by the CAN bit clock, which defines the baud rate on the CAN bus. 1b - The Free Running Timer is clocked by an external time tick. The period can be either adjusted to be equal to the baud rate on the CAN bus, or a different value as required. See the device specific section for details about the external time tick.
14 PREXCEN	Protocol Exception Enable This bit enables the Protocol Exception feature.

Table continues on the next page...

Field	Function
	This field is writable only in Freeze mode. **NOTE:** Refer to Protocol exception event in the CAN Protocol standard (ISO 11898-1) for details. 0b - Protocol Exception is disabled. 1b - Protocol Exception is enabled.
13 —	Reserved
12 ISOCANFDEN	ISO CAN FD Enable This field enables the CAN FD protocol according to ISO specification (ISO 11898-1) (see CAN FD ISO compliance). This field is writable only in Freeze mode. **NOTE:** FlexCAN is able to transmit FD frame format according to CAN Protocol standard (ISO 11898-1). 0b - FlexCAN operates using the non-ISO CAN FD protocol. 1b - FlexCAN operates using the ISO CAN FD protocol (ISO 11898-1).
11 EDFLTDIS	Edge Filter Disable This bit disables the Edge Filter used during the bus integration state. When the Edge Filter is enabled, two consecutive nominal time quanta with dominant bus state are required to detect an edge that causes synchronization. When synchronization occurs, the counting of the sequence of eleven consecutive recessive bits is restarted. The Edge Filter prevents the dominant pulses that are shorter than a nominal bit time (present during the data phase of an FD Frame) from being mistaken for an idle condition. This field is writable only in Freeze mode. **NOTE:** Refer to Bus Integration state in the CAN Protocol standard (ISO 11898-1) for details. 0b - Edge Filter is enabled 1b - Edge Filter is disabled
10 —	Reserved
9-6 —	Reserved
5-0 —	Reserved

37.3.2.13 Error and Status 2 register (ESR2)

37.3.2.13.1 Offset

Register	Offset
ESR2	38h

37.3.2.13.2 Function

This register reports some general status information.

37.3.2.13.3 Diagram

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

37.3.2.13.4 Fields

Field	Function
31-23 —	Reserved
22-16 LPTM	Lowest Priority Tx Mailbox If CAN_ESR2[VPS] is asserted, this field indicates the lowest number inactive Mailbox (see the CAN_ESR2[IMB] bit description). If there is no inactive Mailbox then the Mailbox indicated depends on CAN_CTRL1[LBUF] bit value. If CAN_CTRL1[LBUF] bit is negated then the Mailbox indicated is the one that has the greatest arbitration value (see the "Highest priority Mailbox first" section). If CAN_CTRL1[LBUF] bit is asserted then the Mailbox indicated is the highest number active Tx Mailbox. If a Tx Mailbox is being transmitted it is not considered in LPTM calculation. If CAN_ESR2[IMB] is not asserted and a frame is transmitted successfully, LPTM is updated with its Mailbox number.
15 —	Reserved
14 VPS	Valid Priority Status This bit indicates whether CAN_ESR2[IMB] and CAN_ESR2[LPTM] contents are currently valid or not. It is asserted upon every complete Tx arbitration process unless the CPU writes to Control and Status word of a Mailbox that has already been scanned, that is, it is behind Tx Arbitration Pointer, during the Tx arbitration process. If there is no inactive Mailbox and only one Tx Mailbox that is being transmitted then VPS is not asserted. This bit is negated upon the start of every Tx arbitration process or upon a write to Control and Status word of any Mailbox. **NOTE:** CAN_ESR2[VPS] is not affected by any CPU write into Control Status (C/S) of a MB that is blocked by abort mechanism. When CAN_MCR[AEN] is asserted, the abort code write in C/S of a MB that is being transmitted (pending abort), or any write attempt into a Tx MB with CAN_IFLAG set is blocked. 0b - Contents of IMB and LPTM are invalid. 1b - Contents of IMB and LPTM are valid.
13 IMB	Inactive Mailbox If CAN_ESR2[VPS] is asserted, this bit indicates whether there is any inactive Mailbox (CODE field is either 0b1000 or 0b0000). This bit is asserted in the following cases: • During arbitration, if an CAN_ESR2[LPTM] is found and it is inactive. • If CAN_ESR2[IMB] is not asserted and a frame is transmitted successfully.

Table continues on the next page...

Field	Function
	This bit is cleared in all start of arbitration (see Section "Arbitration process"). **NOTE:** CAN_ESR2[LPTM] mechanism have the following behavior: if an MB is successfully transmitted and CAN_ESR2[IMB]=0 (no inactive Mailbox), then CAN_ESR2[VPS] and CAN_ESR2[IMB] are asserted and the index related to the MB just transmitted is loaded into CAN_ESR2[LPTM]. 0b - If CAN_ESR2[VPS] is asserted, the CAN_ESR2[LPTM] is not an inactive Mailbox. 1b - If CAN_ESR2[VPS] is asserted, there is at least one inactive Mailbox. LPTM content is the number of the first one.
12-0 —	Reserved

37.3.2.14 CRC Register (CRCR)

37.3.2.14.1 Offset

Register	Offset
CRCR	44h

37.3.2.14.2 Function

This register provides information about the CRC of transmitted messages for non FD messages. This register only reports the 15 low order bits of CRC calculations for messages in CAN FD format that require either 17 or 21 bits. For CAN FD format frames, the CAN_FDCRC register must be used. This register is updated at the same time the Tx Interrupt Flag is asserted.

NOTE

Refer to CRC sequence calculation in the CAN Protocol standard (ISO 11898-1) for details.

37.3.2.14.3 Diagram

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

37.3.2.14.4 Fields

Field	Function
31-23 —	Reserved
22-16 MBCRC	CRC Mailbox This field indicates the number of the Mailbox corresponding to the value in CAN_CRCR[TXCRC] field.
15 —	Reserved
14-0 TXCRC	Transmitted CRC value This field indicates the CRC value of the last transmitted message for non-FD frames. For FD frames, CRC value is reported in CAN_FDCRC register.

37.3.2.15 Rx FIFO Global Mask register (RXFGMASK)

37.3.2.15.1 Offset

Register	Offset
RXFGMASK	48h

37.3.2.15.2 Function

This register is located in RAM.

If Rx FIFO is enabled, RXFGMASK is used to mask the Rx FIFO ID Filter Table elements that do not have a corresponding RXIMR according to CAN_CTRL2[RFFN] field setting.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

The following table shows how the FGM bits correspond to each IDAF field.

Table 37-5. Correspondence of Rx FIFO global mask bits to IDF fields

Rx FIFO ID Filter Table Elements Format (CAN_MCR[IDAM])	Identifier Acceptance Filter Fields					
	RTR	IDE	RXIDA	RXIDB [1]	RXIDC [2]	Reserved
A	FGM[31]	FGM[30]	FGM[29:1]	-	-	FGM[0]
B	FGM[31], FGM[15]	FGM[30], FGM[14]	-	FGM[29:16], FGM[13:0]		-
C	-	-		-	FGM[31:24], FGM[23:16], FGM[15:8], FGM[7:0]	

1. If CAN_MCR[IDAM] field is equivalent to the format B only the fourteen most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.
2. If CAN_MCR[IDAM] field is equivalent to the format C only the eight most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.

37.3.2.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FGM															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

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

37.3.2.15.4 Fields

Field	Function
31-0 FGM	Rx FIFO Global Mask Bits These bits mask the ID Filter Table elements bits in a perfect alignment.

Field	Function
	0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

37.3.2.16 Rx FIFO Information Register (RXFIR)

37.3.2.16.1 Offset

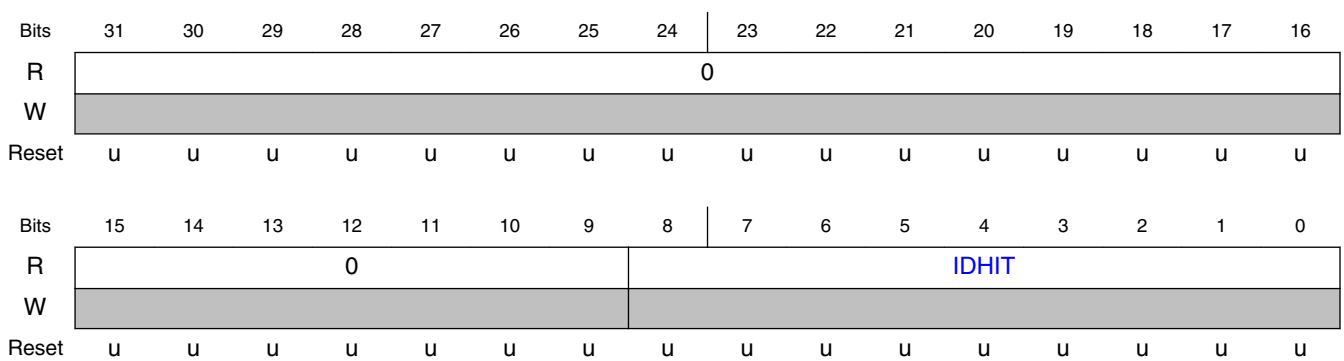
Register	Offset
RXFIR	4Ch

37.3.2.16.2 Function

RXFIR provides information on Rx FIFO.

This register is the port through which the CPU accesses the output of the RXFIR FIFO located in RAM. The RXFIR FIFO is written by the FlexCAN whenever a new message is moved into the Rx FIFO as well as its output is updated whenever the output of the Rx FIFO is updated with the next message. See Section "Rx FIFO" for instructions on reading this register.

37.3.2.16.3 Diagram



37.3.2.16.4 Fields

Field	Function
31-9	Reserved

Table continues on the next page...

Field	Function
—	
8-0 IDHIT	Identifier Acceptance Filter Hit Indicator This field indicates which Identifier Acceptance Filter was hit by the received message that is in the output of the Rx FIFO. If multiple filters match the incoming message ID then the first matching IDAF found (lowest number) by the matching process is indicated. This field is valid only while the CAN_IFLAG1[BUF5I] is asserted.

37.3.2.17 CAN Bit Timing Register (CBT)

37.3.2.17.1 Offset

Register	Offset
CBT	50h

37.3.2.17.2 Function

This register is an alternative way to store the CAN bit timing variables described in CAN_CTRL1 register. EPRESDIV, EPROPSEG, EPSEG1, EPSEG2 and ERJW are extended versions of PRESDIV, PROPSEG, PSEG1, PSEG2 and RJW bit fields respectively.

The BTF bit selects the use of the timing variables defined in this register.

The contents of this register are not affected by soft reset.

NOTE

The CAN bit variables in CAN_CTRL1 and in CAN_CBT are stored in the same register.

NOTE

When the CAN FD feature is enabled (CAN_MCR[FDEN] is set), always set CAN_CBT[BTF].

NOTE

The user must ensure bit time settings and protocol engine tolerance are in compliance with the CAN Protocol standard (ISO 11898-1).

37.3.2.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BTF							EPRESDIV						ERJW		
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	EPROPSEG						EPSEG1					EPSEG2				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

37.3.2.17.4 Fields

Field	Function
31 BTF	Bit Timing Format Enable Enables the use of extended CAN bit timing fields EPRESDIV, EPROPSEG, EPSEG1, EPSEG2 and ERJW replacing the CAN bit timing variables defined in CAN_CTRL1 register. This field can be written in Freeze mode only. 0b - Extended bit time definitions disabled. 1b - Extended bit time definitions enabled.
30-21 EPRESDIV	Extended Prescaler Division Factor This 10-bit field defines the ratio between the PE clock frequency and the Serial Clock (Sclock) frequency when CAN_CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CAN_CTRL1[PRESDIV] value range. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency (see Protocol timing). This field can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (EPRESDIV + 1)
20-16 ERJW	Extended Resync Jump Width This 5-bit field defines the maximum number of time quanta that a bit time can be changed by one re-synchronization when CAN_CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CAN_CTRL1[RJW] value range. One time quantum is equal to the Sclock period. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = ERJW + 1.
15-10 EPROPSEG	Extended Propagation Segment This 6-bit field defines the length of the Propagation Segment in the bit time when CAN_CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CAN_CTRL1[PROPSEG] value range. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation Segment Time = (EPROPSEG + 1) × Time-Quanta. Time-Quantum = one Sclock period.
9-5 EPSEG1	Extended Phase Segment 1

Table continues on the next page...

Field	Function
	This 5-bit field defines the length of Phase Segment 1 in the bit time when CAN_CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CAN_CTRL1[PSEG1] value range. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 1 = (EPSEG1 + 1) × Time-Quanta. Time-Quantum = one Sclock period.
4-0 EPSEG2	Extended Phase Segment 2 This 5-bit field defines the length of Phase Segment 2 in the bit time when CAN_CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CAN_CTRL1[PSEG2] value range. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 1 = (EPSEG2 + 1) × Time-Quanta. Time-Quantum = one Sclock period.

37.3.2.18 Rx Individual Mask Registers (RXIMR0 - RXIMR31)

37.3.2.18.1 Offset

For n = 0 to 31:

Register	Offset
RXIMRn	880h + (n × 4h)

37.3.2.18.2 Function

The RX Individual Mask Registers are used to store the acceptance masks for ID filtering in Rx MBs and the Rx FIFO.

When the Rx FIFO is disabled (CAN_MCR[RFEN] bit is negated), an individual mask is provided for each available Rx Mailbox on a one-to-one correspondence. When the Rx FIFO is enabled (CAN_MCR[RFEN] bit is asserted), an individual mask is provided for each Rx FIFO ID Filter Table Element on a one-to-one correspondence depending on the setting of CAN_CTRL2[RFFN] (see Rx FIFO).

CAN_RXIMR0 stores the individual mask associated to either MB0 or ID Filter Table Element 0, CAN_RXIMR1 stores the individual mask associated to either MB1 or ID Filter Table Element 1 and so on.

CAN_RXIMR registers can only be accessed by the CPU while the module is in Freeze mode, otherwise, they are blocked by hardware. These registers are not affected by reset. They are located in RAM and must be explicitly initialized prior to any reception.

It is possible for the RXIMR memory region to be accessed as general purpose memory. See Bus interface for more information.

37.3.2.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MI															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

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

37.3.2.18.4 Fields

Field	Function
31-0 MI	Individual Mask Bits Each Individual Mask Bit masks the corresponding bit in both the Mailbox filter and Rx FIFO ID Filter Table element in distinct ways. For Mailbox filters, see the RXMGMASK register description. For Rx FIFO ID Filter Table elements, see the RXFGMASK register description. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

37.3.2.19 CAN FD Control Register (FDCTRL)

37.3.2.19.1 Offset

Register	Offset
FDCTRL	C00h

37.3.2.19.2 Function

This register contains control bits for the CAN FD operation. It also defines the data size of Message Buffers allocated in different partitions of RAM (memory blocks) as described in the table below.

When 8 bytes payload is selected:

- Block R0 allocates MB0 to MB31.
- Block R1 allocates MB32 to MB63.

When more than 8 bytes payload is selected, the maximum number of MBs in a block is limited as described below:

Table 37-6. Number of Message Buffers

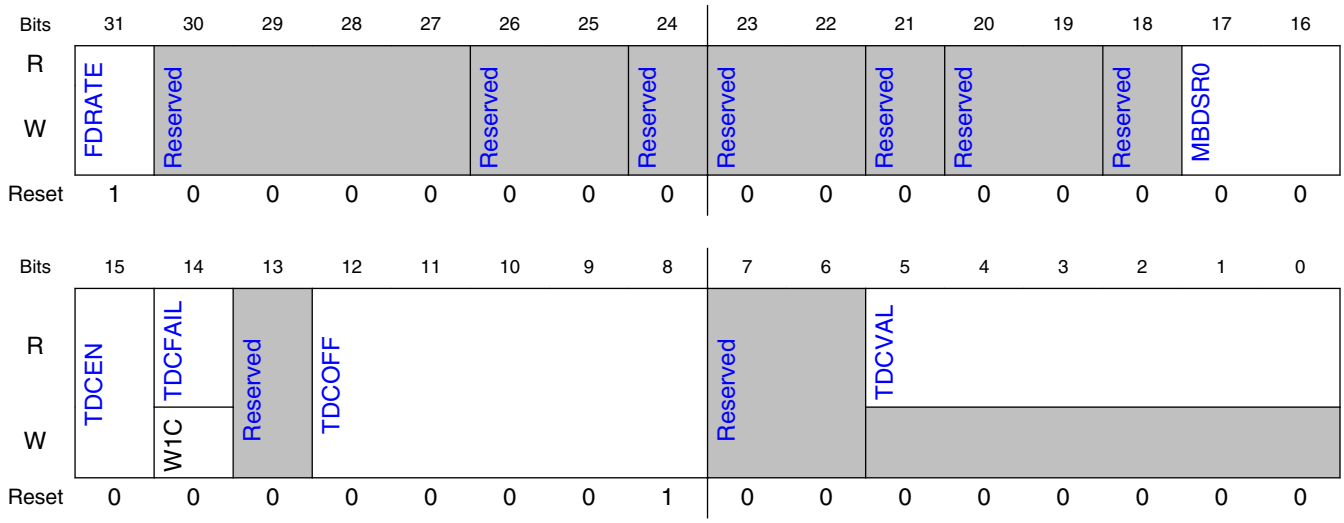
Payload Size	Maximum number of Message Buffers per RAM block
8 bytes	32
16 bytes	21
32 bytes	12
64 bytes	7

NOTE

One memory block fits exactly 32 MBs with 8 bytes payload. For the other options of payload sizes, empty memory may exist between last MB in a block and the beginning of the next block. This empty memory corresponds to less than one MB, and must not be used.

The contents of this register are not affected by soft reset.

37.3.2.19.3 Diagram



37.3.2.19.4 Fields

Field	Function
31 FDRATE	Bit Rate Switch Enable This bit enables the effect of the Bit Rate Switch (BRS bit) during the data phase of Tx messages. The CPU can write this bit any time. However, its effect turns active only when the CAN bus is in Wait for Bus Idle, Bus Idle or Bus Off state, or when the current frame under reception or transmission reaches the interframe space. By negating the CAN_FDCTRL[FDRATE] bit, the CPU can force all bits in CAN FD messages to be transmitted in nominal bit rate, despite of the value in the BRS bit of the Tx MBs. 0b - Transmit a frame in nominal rate. The BRS bit in the Tx MB has no effect. 1b - Transmit a frame with bit rate switching if the BRS bit in the Tx MB is recessive.
30-27 —	Reserved
26-25 —	Reserved
24 —	Reserved
23-22 —	Reserved
21 —	Reserved
20-19 —	Reserved
18 —	Reserved

Table continues on the next page...

Field	Function
17-16 MBDSR0	Message Buffer Data Size for Region 0 This two bit field selects the data size (8, 16, 32 or 64 bytes) for the region R0 of Message Buffers allocated in RAM. It can be written in Freeze Mode only. 00b - Selects 8 bytes per Message Buffer. 01b - Selects 16 bytes per Message Buffer. 10b - Selects 32 bytes per Message Buffer. 11b - Selects 64 bytes per Message Buffer.
15 TDCEN	Transceiver Delay Compensation Enable This bit can be used to enable and disable the TDC feature. It can be written in Freeze mode only. **NOTE:** Refer to Transmitter delay compensation in the CAN Protocol standard (ISO 11898-1) for details. **NOTE:** TDC must be disabled when the Loop Back Mode is enabled (see CAN_CTRL1[LPB] register). 0b - TDC is disabled 1b - TDC is enabled
14 TDCFAIL	Transceiver Delay Compensation Fail This bit indicates when the Transceiver Delay Compensation (TDC) mechanism is out of range, unable to compensate the transceiver's loop delay and successfully compare the delayed received bits to the transmitted ones (see Transceiver Delay Compensation. TDCFAIL sets in the first time FlexCAN detects the out of range condition. The CPU needs to write 1 to clear it. 0b - Measured loop delay is in range. 1b - Measured loop delay is out of range.
13 —	Reserved
12-8 TDCOFF	Transceiver Delay Compensation Offset This bit field contains the offset value to be added to the measured transceiver's loop delay in order to define the position of the delayed comparison point when bit rate switching is active. See Transceiver Delay Compensation for more details on how the loop delay measurement is performed. TDCOFF can be written in Freeze mode only. Its value can be defined in Protocol Engine (PE) Clock periods (CANCLK, see Protocol timing for more details), and must be selected to be smaller than the CAN bit duration in the data bit rate for proper operation. **NOTE:** It is not recommended to use TDCOFF equal to zero.
7-6 —	Reserved
5-0 TDCVAL	Transceiver Delay Compensation Value This register contains the value of the transceiver loop delay measured from the transmitted EDL to R0 transition edge to the respective received one added to the TDCOFF value specified in the CAN_FDCTRL register. This value is an integer multiple of the Protocol Engine (PE) Clock period (CANCLK). See Protocol timing for more details on how the loop delay measurement is performed.

37.3.2.20 CAN FD Bit Timing Register (FDCBT)

37.3.2.20.1 Offset

Register	Offset
FDCBT	C04h

37.3.2.20.2 Function

This register stores the CAN bit timing variables used in the data phase of CAN FD messages when the CAN_FDCTRL[FDRATE] is set, compatible with CAN FD specification. FPRESDIV, FPROPSEG, FPSEG1, FPSEG2 and FRJW are used to define the time quantum duration, the number of time quanta per CAN bit and the sample point position for the data bit rate portion of a CAN FD message with the BRS bit set.

The contents of this register are not affected by soft reset.

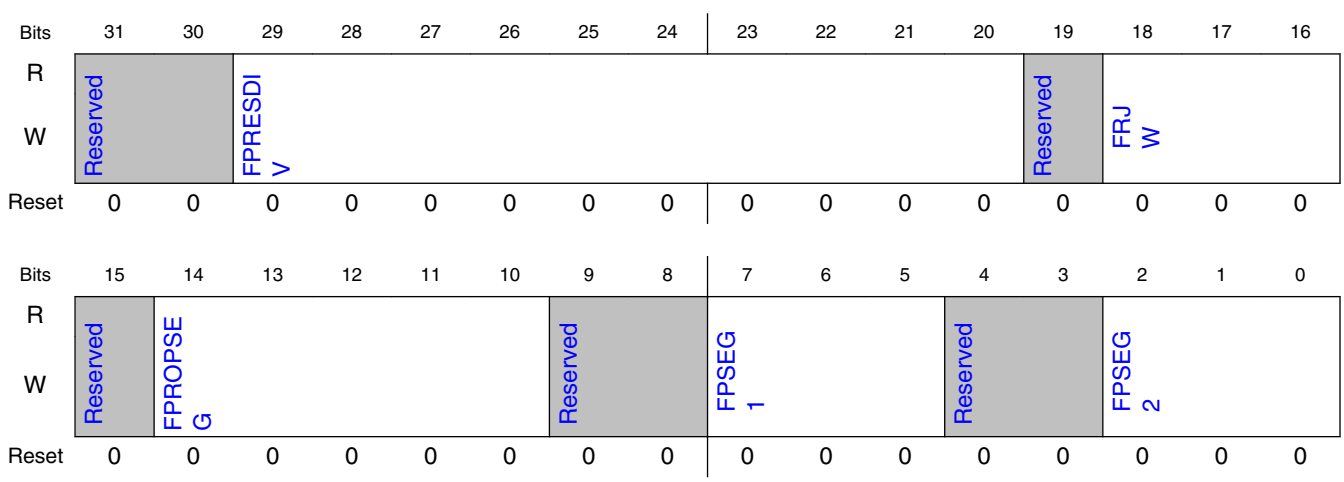
NOTE

The sum of the Fast Propagation Segment (FPROPSEG) and Fast Phase Segment 1 (FPSEG1) must be at least two time quanta.

NOTE

The user must ensure bit time settings and protocol engine tolerance are in compliance with the CAN Protocol standard (ISO 11898-1).

37.3.2.20.3 Diagram



37.3.2.20.4 Fields

Field	Function
31-30 —	Reserved
29-20 FPRESDIV	Fast Prescaler Division Factor This 10-bit field defines the ratio between the PE clock frequency and the Serial Clock (Sclock) frequency in the data bit rate portion of a CAN FD message with the BRS bit set. The Sclock period defines the time quantum of the CAN FD protocol for the data bit rate. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (FPRESDIV + 1). **NOTE:** To minimize errors when processing FD frames, use the same value for FPRESDIV and PRESDIV (in CAN_CBT or CAN_CTRL1). For more details refer to the first NOTE in section CAN FD frames.
19 —	Reserved
18-16 FRJW	Fast Resync Jump Width This 3-bit field defines the maximum number of time quanta that a bit time can be changed by one re-synchronization in the data bit rate portion of a CAN FD message with the BRS bit set. One time quantum is equal to the Sclock period. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = FSJW + 1.
15 —	Reserved
14-10 FPROPSEG	Fast Propagation Segment This 5-bit field defines the length of the Propagation Segment in the bit time in the data bit rate portion of a CAN FD message with the BRS bit set. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation Segment Time = FPROPSEG × Time-Quanta. Time-Quantum = one Sclock period.
9-8 —	Reserved
7-5 FPSEG1	Fast Phase Segment 1 This 3-bit field defines the length of Phase Segment 1 in the bit time in the data bit rate portion of a CAN FD message with the BRS bit set. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Segment 1 = (FPSEG1 + 1) × Time-Quanta. Time-Quantum = one Sclock period.
4-3 —	Reserved
2-0 FPSEG2	Fast Phase Segment 2 This 3-bit field defines the length of Phase Segment 2 in the data bit rate portion of a CAN FD message with the BRS bit set. This field can be written only in Freeze mode because it is blocked by hardware in other modes.

Field	Function
	Phase Segment 2 = (FPSEG2 + 1) × Time-Quanta. Time-Quantum = one Sclock period.

37.3.2.21 CAN FD CRC Register (FDCRC)

37.3.2.21.1 Offset

Register	Offset
FDCRC	C08h

37.3.2.21.2 Function

This register provides information about the CRC of transmitted messages.

FlexCAN uses different CRC polynomials for different frame formats, as shown below.

The CRC_15 polynomial is used for all frames in CAN format. The CRC_17 polynomial is used for frames in CAN FD format with a DATA FIELD up to sixteen bytes. The CRC_21 polynomial is used for frames in CAN FD format with a DATA FIELD longer than sixteen bytes. Each polynomial shown below results in a Hamming Distance of 6. This register is updated at the same time the Tx Interrupt Flag is asserted.

CRC_15 = 0xC599: $(x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1)$

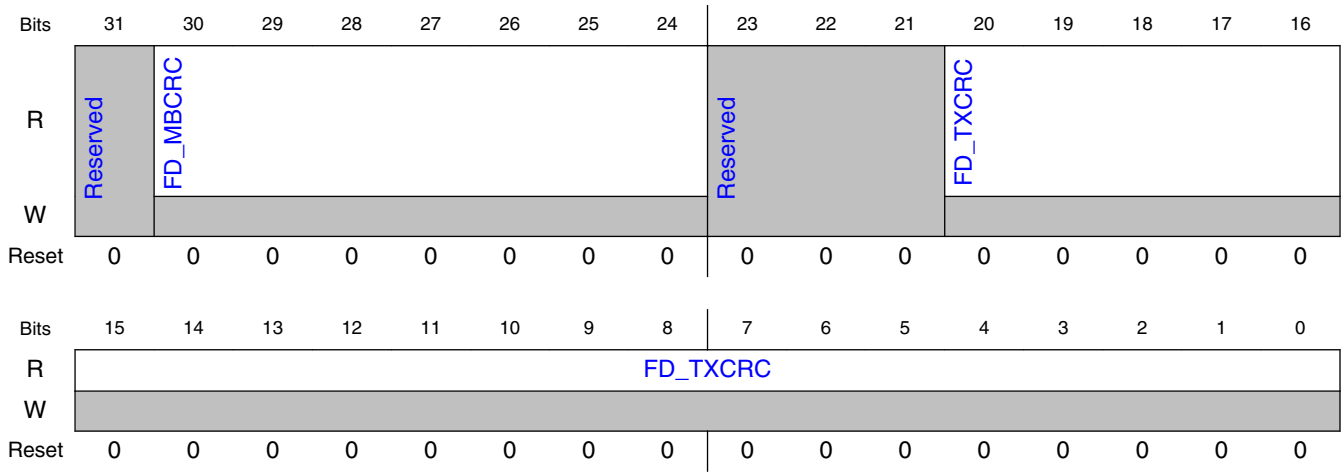
CRC_17 = 0x3685B: $(x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^6 + x^4 + x^3 + x^1 + 1)$

CRC_21 = 0x302899: $(x^{21} + x^{20} + x^{13} + x^{11} + x^7 + x^4 + x^3 + 1)$

NOTE

Refer to CRC sequence calculation in the CAN Protocol standard (ISO 11898-1) for details.

37.3.2.21.3 Diagram



37.3.2.21.4 Fields

Field	Function
31 —	Reserved
30-24 FD_MBCRC	CRC Mailbox Number for FD_TXCRC This field indicates the number of the Mailbox corresponding to the value in FD_TXCRC field, for both FD and non-FD frames. It reports the same information as in MBCRC bit field in CAN_CRCR register.
23-21 —	Reserved
20-0 FD_TXCRC	Extended Transmitted CRC value This 21-bit field contains the CRC value calculated over the most recent transmitted message. Different CRC polynomials are used for different frame formats. A 15-bit polynomial, CRC_15, is used for all frames in CAN format. The second 17-bit polynomial, CRC_17, is used for frames in CAN FD format with a data field up to sixteen bytes long. The third 21-bit polynomial, CRC_21, is used for frames in CAN FD format with a data field longer than sixteen bytes. For CRC_15 and CRC_17, the 6 most significant bits and the 4 most significant bits are reported as zeros, respectively. For CRC_15, this register has the same content as CRC Register.

37.3.3 Message buffer structure

The message buffer structure used by the FlexCAN module is represented in the following figure. Both Extended (29-bit identifier) and Standard (11-bit identifier) frames used in the CAN specification (Version 2.0 Part B) are represented. Each individual MB is formed by 16, 24, 40 or 72 bytes, depending on the quantity of data bytes allocated for the message payload: 8, 16, 32 or 64 data bytes, respectively.

The memory area from 0x80 to 0x27F is used by the mailboxes. When CAN FD is enabled, the exact address for each MB depends on the size of its payload. See FlexCAN Memory Partition for CAN FD for more detailed information.

Table 37-7. Message buffer structure - example with 64 bytes payload

	31	30	29	28	27	24	23	22	21	20	19	18	17	16	15	8	7	0
0x0	EDL	BRS	ESI		CODE			SRR	IDE	RTR	DLC				TIME STAMP			
0x4	PRIO				ID (Standard/Extended)										ID (Extended)			
0x8	Data Byte 0							Data Byte 1							Data Byte 2		Data Byte 3	
0xC	Data Byte 4							Data Byte 5							Data Byte 6		Data Byte 7	
0x10	Data Byte 8							Data Byte 9							Data Byte 10		Data Byte 11	
0x14	Data Byte 12							Data Byte 13							Data Byte 14		Data Byte 15	
0x18	Data Byte 16							Data Byte 17							Data Byte 18		Data Byte 19	
0x1C	Data Byte 20							Data Byte 21							Data Byte 22		Data Byte 23	
0x20	Data Byte 24							Data Byte 25							Data Byte 26		Data Byte 27	
0x24	Data Byte 28							Data Byte 29							Data Byte 30		Data Byte 31	
0x28	Data Byte 32							Data Byte 33							Data Byte 34		Data Byte 35	
0x2C	Data Byte 36							Data Byte 37							Data Byte 38		Data Byte 39	
0x30	Data Byte 40							Data Byte 41							Data Byte 42		Data Byte 43	
0x34	Data Byte 44							Data Byte 45							Data Byte 46		Data Byte 47	
0x38	Data Byte 48							Data Byte 49							Data Byte 50		Data Byte 51	
0x3C	Data Byte 52							Data Byte 53							Data Byte 54		Data Byte 55	
0x40	Data Byte 56							Data Byte 57							Data Byte 58		Data Byte 59	
0x44	Data Byte 60							Data Byte 61							Data Byte 62		Data Byte 63	

= Unimplemented or Reserved

EDL - Extended Data Length

This bit distinguishes between CAN format and CAN FD format frames. The EDL bit must not be set for Message Buffers configured to RANSWER with code field 0b1010 (see Table below).

BRS - Bit Rate Switch

This bit defines whether the bit rate is switched inside a CAN FD format frame.

ESI - Error State Indicator

This bit indicates if the transmitting node is error active or error passive.

CODE - Message Buffer Code

This 4-bit field can be accessed (read or write) by the CPU and by the FlexCAN module itself, as part of the message buffer matching and arbitration process. The encoding is shown in Table 37-8 and Table 37-9. See Functional description for additional information.

Table 37-8. Message buffer code for Rx buffers

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
0b0000: INACTIVE - MB is not active.	INACTIVE	-	-	-	MB does not participate in the matching process.
0b0100: EMPTY - MB is active and empty.	EMPTY	-	FULL	-	When a frame is received successfully (after the Move-in) process), the CODE field is automatically updated to FULL.
0b0010: FULL - MB is full.	FULL	Yes	FULL	-	The act of reading the C/S word followed by unlocking the MB (SRV) does not make the code return to EMPTY. It remains FULL. If a new frame is moved to the MB after the MB was serviced, the code still remains FULL. See Matching process for matching details related to FULL code.
		No	OVERRUN	-	If the MB is FULL and a new frame is moved to this MB before the CPU services it, the CODE field is automatically updated to OVERRUN. See Matching process

Table continues on the next page...

Table 37-8. Message buffer code for Rx buffers (continued)

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
					for details about overrun behavior.
0b0110: OVERRUN - MB is being overwritten into a full buffer.	OVERRUN	Yes	FULL	-	If the CODE field indicates OVERRUN and CPU has serviced the MB, when a new frame is moved to the MB then the code returns to FULL.
		No	OVERRUN	-	If the CODE field already indicates OVERRUN, and another new frame must be moved, the MB will be overwritten again, and the code will remain OVERRUN. See Matching process for details about overrun behavior.
0b1010: RANSWER[4] - A frame was configured to recognize a Remote Request Frame and transmit a Response Frame in return.[5]	RANSWER	-	TANSWER(0b1110)	0	A Remote Answer was configured to recognize a remote request frame received. After that an MB is set to transmit a response frame. The code is automatically changed to TANSWER (0b1110). See Matching process for details. If CAN_CTRL2[RRS] is negated, transmit a response frame whenever a remote request frame with the same ID is received.
		-	-	1	This code is ignored during matching and arbitration process. See Matching process for details.

Table continues on the next page...

Table 37-8. Message buffer code for Rx buffers (continued)

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
CODE[0]=1: BUSY - FlexCAN is updating the contents of the MB. The CPU must not access the MB.	BUSY[6]	-	FULL	-	Indicates that the MB is being updated. It will be negated automatically and does not interfere with the next CODE.
		-	OVERRUN	-	

1. SRV: Serviced MB. MB was read and unlocked by reading TIMER or other MB.
2. A frame is considered a successful reception after the frame to be moved to MB (move-in process). See Move-in for details.
3. Remote Request Stored bit, see "Control 2 Register (CAN_CTRL2)" for details.
4. Code 0b1010 is not considered Tx and an MB with this code should not be aborted.
5. Code 0b1010 must be used in Message Buffers configured in CAN FD format, having the EDL bit set.
6. Note that for Tx MBs, the BUSY bit should be ignored upon read, except when AEN bit is set in the MCR register. If this bit is asserted, the corresponding MB does not participate in the matching process.

Table 37-9. Message buffer code for Tx buffers

CODE Description	Tx Code BEFORE tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
0b1000: INACTIVE - MB is not active	INACTIVE	-	-	MB does not participate in arbitration process.
0b1001: ABORT - MB is aborted	ABORT	-	-	MB does not participate in arbitration process.
0b1100: DATA - MB is a Tx Data Frame (MB RTR must be 0)	DATA	0	INACTIVE	Transmit data frame unconditionally once. After transmission, the MB automatically returns to the INACTIVE state.
0b1100: REMOTE - MB is a Tx Remote Request Frame (MB RTR must be 1)	REMOTE	1	EMPTY	Transmit remote request frame unconditionally once. After transmission, the MB automatically becomes an Rx Empty MB with the same ID.
0b1110: TANSWER - MB is a Tx Response Frame from an incoming Remote Request Frame	TANSWER	-	RANSWER	This is an intermediate code that is automatically written to the MB by the CHI as a result of a match to a remote request frame. The remote response frame will be transmitted unconditionally once,

Table 37-9. Message buffer code for Tx buffers

CODE Description	Tx Code BEFORE tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
				and then the code will automatically return to RANSWER (0b1010). The CPU can also write this code with the same effect. The remote response frame can be either a data frame or another remote request frame depending on the RTR bit value. See Matching process and Arbitration process for details.

SRR - Substitute Remote Request

Fixed recessive bit, used only in extended format. It must be set to one by the user for transmission (Tx Buffers) and will be stored with the value received on the CAN bus for Rx receiving buffers. It can be received as either recessive or dominant. If FlexCAN receives this bit as dominant, then it is interpreted as an arbitration loss.

1 = Recessive value is compulsory for transmission in extended format frames

0 = Dominant is not a valid value for transmission in extended format frames

IDE - ID Extended Bit

This field identifies whether the frame format is standard or extended.

1 = Frame format is extended

0 = Frame format is standard

RTR - Remote Transmission Request

This bit affects the behavior of remote frames and is part of the reception filter. See Table 37-8, Table 37-9, and the description of the RRS bit in Control 2 Register (CAN_CTRL2) for additional details.

If FlexCAN transmits this bit as '1' (recessive) and receives it as '0' (dominant), it is interpreted as an arbitration loss. If this bit is transmitted as '0' (dominant), then if it is received as '1' (recessive), the FlexCAN module treats it as a bit error. If the value received matches the value transmitted, it is considered a successful bit transmission.

1 = Indicates the current MB may have a remote request frame to be transmitted if MB is Tx. If the MB is Rx then incoming remote request frames may be stored.

0 = Indicates the current MB has a data frame to be transmitted. In Rx MB it may be considered in matching processes.

NOTE

When configuring CAN FD frames, the RTR bit must be negated.

DLC - Length of Data in Bytes

This 4-bit field is the length (in bytes) of the Rx or Tx data, which is located in offset 0x8 through 0xF of the MB space (see Table 37-7). In reception, this field is written by the FlexCAN module, copied from the DLC (Data Length Code) field of the received frame. In transmission, this field is written by the CPU and corresponds to the DLC field value of the frame to be transmitted. When RTR = 1, the frame to be transmitted is a remote frame and does not include the data field, regardless of the DLC field (see Table 37-10).

TIME STAMP - Free-Running Counter Time Stamp

This 16-bit field is a copy of the Free-Running Timer, captured for Tx and Rx frames at the time when the beginning of the Identifier field appears on the CAN bus.

PRIO - Local priority

This 3-bit field is used only when LPRIO_EN bit is set in CAN_MCR, and it only makes sense for Tx mailboxes. These bits are not transmitted. They are appended to the regular ID to define the transmission priority. See Arbitration process.

ID - Frame Identifier

In standard frame format, only the 11 most significant bits (28 to 18) are used for frame identification in both receive and transmit cases. The 18 least significant bits are ignored. In extended frame format, all bits are used for frame identification in both receive and transmit cases.

DATA BYTE 0 to 63 - Data Field

Up to sixty four bytes can be used for a data frame, depending on the size of payload selected for the Message Buffers.

For Rx frames, the data is stored as it is received from the CAN bus. DATA BYTE (*n*) is valid only if *n* is less than DLC as shown in the table below.

Table 37-10. DATA BYTEs validity

DLC	Valid DATA BYTEs
0	none
1	DATA BYTE 0
2	DATA BYTE 0 to 1
3	DATA BYTE 0 to 2
4	DATA BYTE 0 to 3
5	DATA BYTE 0 to 4
6	DATA BYTE 0 to 5
7	DATA BYTE 0 to 6
8	DATA BYTE 0 to 7
9	DATA BYTE 0 to 11
10	DATA BYTE 0 to 15
11	DATA BYTE 0 to 19
12	DATA BYTE 0 to 23
13	DATA BYTE 0 to 31
14	DATA BYTE 0 to 47
15	DATA BYTE 0 to 63

37.3.4 FlexCAN Memory Partition for CAN FD

When CAN FD is enabled, the FlexCAN RAM can be partitioned in blocks of 512 bytes. Each block can accommodate a number of Message Buffers which depends on the configuration provided by CAN_FDCTRL[MBDSRn] bit fields as shown in table below.

Table 37-11. RAM partition

RAM block	Number of MBs with 8 bytes (default range)	Size control bit field in CAN_FDCTRL register	Number of MBs of different sizes, per block
0	0 to 31	MBDSR0	MBDSR0=00, 32 MBs with 8 bytes payload MBDSR0=01, 21 MBs with 16 bytes payload MBDSR0=10, 12 MBs with 32 bytes payload MBDSR0=11, 7 MBs with 64 bytes payload

When payload sizes of 16, 32 or 64 bytes are configured in some or all RAM blocks, the total number of MBs and its respective number order may differ from the default configuration of 8 bytes. For example, suppose Block0 is configured to 8 bytes payload, Block1 to 16 bytes, than the following table indicates how the Message Buffers will be arranged into RAM.

Table 37-12. RAM partition example

RAM block	Payload size	Number of MBs in the RAM block	Message Buffer range
0	CAN_FDCTRL[MBDSR0]=00, 8 bytes payload	32	0 to 31

37.3.5 FlexCAN message buffer memory map

The FlexCAN memory buffers are allocated in memory according to the tables below.

Table 37-13. 8-byte message buffers

Address offset (hex)	MBDSR=b00 8-byte payload
0080	MB0
0090	MB1
00A0	MB2
00B0	MB3
00C0	MB4
00D0	MB5
00E0	MB6
00F0	MB7
0100	MB8
0110	MB9
0120	MB10
0130	MB11
0140	MB12
0150	MB13
0160	MB14
0170	MB15
0180	MB16
0190	MB17
01A0	MB18
01B0	MB19

Table continues on the next page...

Table 37-13. 8-byte message buffers (continued)

Address offset (hex)	MBDSR=b00 8-byte payload
01C0	MB20
01D0	MB21
01E0	MB22
01F0	MB23
0200	MB24
0210	MB25
0220	MB26
0230	MB27
0240	MB28
0250	MB29
0260	MB30
0270	MB31

Table 37-14. 16-byte message buffers

Address offset (hex)	MBDSR=b01 16-byte payload
0080	MB0
0098	MB1
00B0	MB2
00C8	MB3
00E0	MB4
00F8	MB5
0110	MB6
0128	MB7
0140	MB8
0158	MB9
0170	MB10
0188	MB11
01A0	MB12
01B8	MB13
01D0	MB14
01E8	MB15
0200	MB16
0218	MB17
0230	MB18
0248	MB19
0260	MB20

Table 37-15. 32-byte message buffers

Address offset (hex)	MBDSR=b10 32-byte payload
0080	MB0
00A8	MB1
00D0	MB2
00F8	MB3
0120	MB4
0148	MB5
0170	MB6
0198	MB7
01C0	MB8
01E8	MB9
0210	MB10
0238	MB11

Table 37-16. 64-byte message buffers

Address offset (hex)	MBDSR=b11 64-byte payload
0080	MB0
00C8	MB1
0110	MB2
0158	MB3
01A0	MB4
01E8	MB5
0230	MB6

37.3.6 Rx FIFO structure

37.4 Functional description

The FlexCAN module is a CAN protocol engine with a very flexible mailbox system for transmitting and receiving CAN frames. The mailbox system is composed by a set of Message Buffers (MB) that store configuration and control data, time stamp, message ID and data (see Message buffer structure).

For Classical CAN frames, simultaneous reception through FIFO and mailbox is supported. For CAN FD frames, reception is supported through mailboxes only. For mailbox reception, a matching algorithm makes it possible to store received frames only into MBs that have the same ID programmed on its ID field. A masking scheme makes it possible to match the ID programmed on the MB with a range of IDs on received CAN frames. For transmission, an arbitration algorithm decides the prioritization of MBs to be transmitted based on the message ID (optionally augmented by 3 local priority bits) or the MB ordering.

Before proceeding with the functional description, an important concept must be explained. A Message Buffer is said to be "active" at a given time if it can participate in both the Matching and Arbitration processes. An Rx MB with a 0b0000 code is inactive (refer to Table 37-8). Similarly, a Tx MB with a 0b1000 or 0b1001 code is also inactive (refer to Table 37-9).

The FlexCAN module is also able to receive and transmit messages in CAN FD format. The Message Buffers are sized to adequately store the quantity of data bytes selected by the MBDSRn bit fields in CAN_FDCTRL register. The quantity of FD MBs available for a given quantity of data bytes is described CAN_FDCTRL register. See also FlexCAN Memory Partition for CAN FD.

37.4.1 Transmit process

To transmit a CAN frame, the CPU must prepare a Message Buffer for transmission by executing the following procedure:

1. Check whether the respective interrupt bit is set and clear it.
2. If the MB is active (transmission pending), write the ABORT code (0b1001) to the CODE field of the Control and Status word to request an abort of the transmission.
3. Wait for the corresponding IFLAG bit to be asserted by polling the CAN_IFLAG register, or by the interrupt request if enabled by the respective IMASK bit.
4. Read back the CODE field to check if the transmission was aborted or transmitted (see Transmission abort mechanism).
5. Clear the corresponding interrupt flag.
6. Write the ID register (containing the local priority if enabled via MCR[LPRIO_EN]).
7. Write payload data bytes.
8. Configure the Control and Status word with the desired configuration.
   - Set ID type via MB_CS[IDE].
   - Set Remote Transmission Request (if needed) via MB_CS[RTR].
   - If CAN_MCR[FDEN] is enabled, also configure the MB_CS[EDL] and MB_CS[BRS] fields. For details about the relationship between the written value and transmitted value of the MB_CS[ESI] field, see Table 37-20.[1]

- Activate the message buffer to transmit the CAN frame by setting MB_CS[CODE] to 0xC.
- Set Data Length Code in bytes via MB_CS[DLC]. See Table 37-10 for detailed information.

When the MB is activated, it participates in the arbitration process and is eventually transmitted according to its priority. When the DLC value stored in the MB selected for transmission is larger than the respective MB payload size, FlexCAN adds the necessary number of bytes with constant 0xCC pattern to complete the expected DLC.

At the end of the successful transmission:

- The value of the Free Running Timer is written into the Time Stamp field.
- The CODE field in the Control and Status word is updated.
- Both CAN_CRC and CAN_FDCRC registers are updated.
- A status flag is set in the Interrupt Flag register.
- An interrupt is generated if allowed by the corresponding Interrupt Mask Register bit.

The new CODE field after transmission depends on the code that was used to activate the MB (see Table 37-8 and Table 37-9 in Message buffer structure).

When the Abort feature is enabled (CAN_MCR[AEN] is asserted), after the Interrupt Flag is asserted for a Mailbox configured as transmit buffer, the Mailbox is blocked. Therefore the CPU is not able to update it until the Interrupt Flag is negated by the CPU. This means that the CPU must clear the corresponding IFLAG bit before starting to prepare this MB for a new transmission or reception.

NOTE

If backwards compatibility is desired (CAN_MCR[AEN] bit is negated), write the INACTIVE code (0b1000) to the CODE field to inactivate the MB. However, in this case the pending frame may be transmitted without notification (see Mailbox inactivation).

37.4.2 Arbitration process

The arbitration process scans the Mailboxes searching the Tx one that holds the message to be sent in the next opportunity. This Mailbox is called the *arbitration winner*.

1. The ESI field does not need to be written, and will automatically be transmitted dominant by error active nodes and recessive by error passive nodes. Note that there is an exception if the FlexCAN is operating as a network gateway: In that case, the CPU writes the MB_CS[ESI] bit according to the error status of the node which sent the message.

The scan starts from the lowest number Mailbox and runs toward the higher ones.

The arbitration process is triggered in the following events:

- From the CRC field of the CAN frame. The start point depends on the CAN_CTRL2[TASD] field value.
- During the Error Delimiter field of a CAN frame.
- During the Overload Delimiter field of a CAN frame.
- When the winner is inactivated and the CAN bus has still not reached the first bit of the Intermission field.
- When there is CPU write to the C/S word of a winner MB and the CAN bus has still not reached the first bit of the Intermission field.
- When CHI is in Idle state and the CPU writes to the C/S word of any MB.
- When FlexCAN exits Bus Off state.
- Upon leaving Freeze mode or Low Power mode.

If the arbitration process does not manage to evaluate all Mailboxes before the CAN bus has reached the first bit of the Intermission field the temporary arbitration winner is invalidated and the FlexCAN will not compete for the CAN bus in the next opportunity.

The arbitration process selects the winner among the active Tx Mailboxes at the end of the scan according to both CAN_CTRL1[LBUF] and CAN_MCR[LPRIOEN] bits settings.

37.4.2.1 Lowest-number Mailbox first

If CAN_CTRL1[LBUF] bit is asserted the first (lowest number) active Tx Mailbox found is the arbitration winner. CAN_MCR[LPRIOEN] bit has no effect when CAN_CTRL1[LBUF] is asserted.

37.4.2.2 Highest-priority Mailbox first

If CAN_CTRL1[LBUF] bit is negated, then the arbitration process searches the active Tx Mailbox with the highest priority, which means that this Mailbox's frame would have a higher probability to win the arbitration on CAN bus when multiple external nodes compete for the bus at the same time.

The sequence of bits considered for this arbitration is called the *arbitration value* of the Mailbox. The highest-priority Tx Mailbox is the one that has the lowest arbitration value among all Tx Mailboxes.

If two or more Mailboxes have equivalent arbitration values, the Mailbox with the lowest number is the arbitration winner.

The composition of the arbitration value depends on CAN_MCR[LPRIOEN] bit setting.

37.4.2.2.1 Local Priority disabled

If CAN_MCR[LPRIOEN] bit is negated the arbitration value is built in the exact sequence of bits as they would be transmitted in a CAN frame (see the following table) in such a way that the Local Priority is disabled.

Table 37-17. Composition of the arbitration value when Local Priority is disabled

Format	Mailbox Arbitration Value (32 bits)				
Standard (IDE = 0)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bit)
Extended (IDE = 1)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

37.4.2.2.2 Local Priority enabled

If Local Priority is desired CAN_MCR[LPRIOEN] must be asserted. In this case the Mailbox PRIO field is included at the very left of the arbitration value (see the following table).

Table 37-18. Composition of the arbitration value when Local Priority is enabled

Format	Mailbox Arbitration Value (35 bits)					
Standard (IDE = 0)	PRIO (3 bits)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bit)
Extended (IDE = 1)	PRIO (3 bits)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

As the PRIO field is the most significant part of the arbitration value Mailboxes with low PRIO values have higher priority than Mailboxes with high PRIO values regardless the rest of their arbitration values.

Note that the PRIO field is not part of the frame on the CAN bus. Its purpose is only to affect the internal arbitration process.

37.4.2.3 Arbitration process (continued)

After the arbitration winner is found, its content is copied to a hidden auxiliary MB called Tx Serial Message Buffer (Tx SMB), which has the same structure as a normal MB but is not user accessible. This operation is called move-out and after it is done, write access to the C/S word of the corresponding MB is blocked (if the AEN bit in CAN_MCR register is asserted). Write access is restored in the following events:

- After the MB is transmitted and the corresponding IFLAG bit is cleared by the CPU
- FlexCAN enters in Freeze mode or Bus Off
- FlexCAN loses the bus arbitration or there is an error during the transmission

At the first opportunity window on the CAN bus, the message on the Tx SMB is transmitted according to the CAN protocol rules.

Arbitration process can be triggered in the following situations:

- During Rx and Tx frames from CAN CRC field to end of frame. CAN_CTRL2[TASD] bit value may be changed to optimize the arbitration start point.
- During CAN BusOff state from TX_ERR_CNT=124 to 128. CAN_CTRL2[TASD] bit value may be changed to optimize the arbitration start point.
- During C/S write by CPU in BusIdle. First C/S write starts arbitration process and a second C/S write during this same arbitration restarts the process. If other C/S writes are performed, Tx arbitration process is pending. If there is no arbitration winner after the arbitration process has finished, then the TX arbitration machine begins a new arbitration process. If there is a pending arbitration and BusIdle state starts then an arbitration process is triggered. In this case the first and second C/S write in BusIdle will not restart the arbitration process. It is possible that there is not enough time to finish arbitration in WaitForBusIdle state and the next state is Idle. In this case the scan is not interrupted, and it is completed during BusIdle state. During this arbitration C/S write does not cause arbitration restart.
- Arbitration winner deactivation during a valid arbitration window.
- Upon exiting Freeze mode (first bit of the WaitForBusIdle state). If there is a re-synchronization during WaitForBusIdle, the arbitration process is restarted.

Arbitration process stops in the following situations:

- All Mailboxes were scanned
- A Tx active Mailbox is found in case of Lowest Buffer feature enabled
- Arbitration winner inactivation or abort during any arbitration process

- There was not enough time to finish Tx arbitration process (for instance, when a deactivation was performed near the end of frame). In this case arbitration process is pending.
- Error or Overload flag in the bus
- Low Power or Freeze mode request in Idle state

Arbitration is considered pending as described below:

- It was not possible to finish arbitration process in time
- C/S write during arbitration if write is performed in a MB whose number is lower than the Tx arbitration pointer
- Any C/S write if there is no Tx Arbitration process in progress
- Rx Match has just updated a Rx Code to Tx Code
- Entering Busoff state

C/S write during arbitration has the following effect:

- If C/S write is performed in the arbitration winner, a new process is restarted immediately.
- If C/S write is performed in a MB whose number is higher than the Tx arbitration pointer, the ongoing arbitration process will scan this MB as normal.

37.4.3 Receive process

To be able to receive CAN frames into a Mailbox, the CPU must prepare it for reception by executing the following steps:

1. If the Mailbox is active (either Tx or Rx) inactivate the Mailbox (see Mailbox inactivation), preferably with a safe inactivation (see Transmission abort mechanism).
2. Write the ID word
3. Write the EMPTY code (0b0100) to the CODE field of the Control and Status word to activate the Mailbox. No setup is required for EDL, BRS and ESI bits, they are overwritten by the respective bit fields in the received message.

After the MB is activated, it will be able to receive frames that match the programmed filter. At the end of a successful reception, the Mailbox is updated by the *move-in* process (see Move-in) as follows:

1. The received Data field (8 bytes at most for Classical CAN message format and up to 64 bytes for CAN FD message format) is stored.

2. The received Identifier field is stored.
3. The value of the Free Running Timer at the time of the second bit of frame's Identifier field is written into the Mailbox's Time Stamp field.
4. The received SRR, IDE, RTR, EDL, BRS, ESI and DLC fields are stored.
5. The CODE field in the Control and Status word is updated (see Table 37-8 and Table 37-9 in Section Message buffer structure).
6. A status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit.

The recommended way for CPU servicing (read) the frame received in an Mailbox is using the following procedure:

1. Read the Control and Status word of that Mailbox.
2. Check if the BUSY bit is deasserted, indicating that the Mailbox is locked. Repeat step 1) while it is asserted. See Mailbox lock mechanism.
3. Read the contents of the Mailbox. Once Mailbox is locked now, its contents won't be modified by FlexCAN Move-in processes. See Move-in.
4. Acknowledge the proper flag at IFLAG registers.
5. Read the Free Running Timer. It is optional but recommended to unlock Mailbox as soon as possible and make it available for reception.

The CPU should poll for frame reception by the status flag bit for the specific Mailbox in one of the IFLAG Registers and not by the CODE field of that Mailbox. Polling the CODE field does not work because once a frame was received and the CPU services the Mailbox (by reading the C/S word followed by unlocking the Mailbox), the CODE field will not return to EMPTY. It will remain FULL, as explained in Table 37-8. If the CPU tries to workaround this behavior by writing to the C/S word to force an EMPTY code after reading the Mailbox without a prior *safe inactivation*, a newly received frame matching the filter of that Mailbox may be lost.

CAUTION

In summary: *never do polling by reading directly the C/S word of the Mailboxes. Instead, read the IFLAG registers.*

Note that the received frame's Identifier field is always stored in the matching Mailbox, thus the contents of the ID field in an Mailbox may change if the match was due to masking. When CAN_MCR[SRXDIS] bit is asserted, FlexCAN will not store frames transmitted by itself in any MB, even if it contains a matching Rx Mailbox, and no

interrupt flag or interrupt signal will be generated. Otherwise, when CAN_MCR[SRXDIS] bit is deasserted, FlexCAN can receive frames transmitted by itself if there exists a matching Rx Mailbox.

When CAN_MCR[DMA] is asserted, upon receiving a frame in FIFO, CAN_IFLAG1[BUF5I] generates a DMA request and does not generate a CPU interrupt (see Rx FIFO under DMA Operation). The CAN_IMASK1 bits in Rx FIFO region are not used.

The DMA controller must service the received frame using the following procedure:

1. Read the Control and Status word (read 0x80 address, optional)
2. Read the ID field (read 0x84 address, optional)
3. Read all Data Bytes (start read at 0x88 address, optional)
4.

37.4.4 Matching process

The matching process scans the MB memory looking for Rx MBs programmed with the same ID as the one received from the CAN bus. . The matching starts from the lowest number Message Buffer toward the higher ones.

As the frame is being received, it is stored in a hidden auxiliary MB called Rx Serial Message Buffer (Rx SMB).

The matching process start point depends on the following conditions:

- If the received frame is a remote frame, the start point is the CRC field of the frame
- If the received frame is a data frame with DLC field equal to zero, the start point is the CRC field of the frame
- If the received frame is a data frame with DLC field different than zero, the start point is the DATA field of the frame

If a matching ID is found in one of the Mailboxes, the contents of the Rx SMB are transferred to to the matched Mailbox by the move-in process. If any CAN protocol error is detected then no match results are transferred to the matched Mailbox at the end of reception.

The matching process scans all matching elements of the active Rx Mailboxes (CODE is EMPTY, FULL, OVERRUN or RANSWER) in search of a successful comparison with the matching elements of the Rx SMB that is receiving the frame on the CAN bus. The

Rx SMB has the same structure of a Mailbox. The reception structures (Mailboxes) associated with the matching elements that had a successful comparison are the *matched structures*. The *matching winner* is selected at the end of the scan among those matched structures and depends on conditions described ahead. See the following table.

Table 37-19. Matching architecture

Structure	SMB[RTR]	CTRL2[RRS]	CTRL2[EACEN]	MB[IDE]	MB[RTR]	MB[ID[1]]	MB[CODE]
Mailbox	0	-	0	cmp[2]	no_cmp[3]	cmp_msk[4]	EMPTY or FULL or OVERRUN
Mailbox	0	-	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN
Mailbox	1	0	-	cmp	no_cmp	cmp	RANSWER
Mailbox	1	1	0	cmp	no_cmp	cmp_msk	EMPTY or FULL or OVERRUN
Mailbox	1	1	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN

1. For Mailbox structure, If SMB[IDE] is asserted, the ID is 29 bits (ID Standard + ID Extended). If SMB[IDE] is negated, the ID is only 11 bits (ID Standard). For FIFO structure, the ID depends on IDAM.
2. cmp: Compares the Rx SMB contents with the MB contents regardless the masks.
3. no_cmp: The Rx SMB contents are not compared with the MB contents.
4. cmp_msk: Compares the Rx SMB contents with MB contents taking into account the masks.

A reception structure is *free-to-receive* when any of the following conditions is satisfied:

- The CODE field of the Mailbox is EMPTY
- The CODE field of the Mailbox is either FULL or OVERRUN and it has already been serviced (the C/S word was read by the CPU and unlocked as described in Mailbox lock mechanism)
- The CODE field of the Mailbox is either FULL or OVERRUN and an inactivation (see Mailbox inactivation) is performed

The scan order for Mailboxes is from the matching element with lowest number to the higher ones.

The matching winner search for Mailboxes is affected by the CAN_MCR[IRMQ] bit. If it is negated, the matching winner is the first matched Mailbox regardless if it is free-to-receive or not. If it is asserted, the matching winner is selected according to the priority below:

1. the first free-to-receive matched Mailbox;
2. the last non free-to-receive matched Mailbox.

If a non-safe Mailbox inactivation (see Mailbox inactivation) occurs during matching process and the Mailbox inactivated is the temporary matching winner, then the temporary matching winner is invalidated. The matching elements scan is not stopped nor restarted, it continues normally. The consequence is that the current matching process works as if the matching elements compared before the inactivation did not exist, therefore a message may be lost.

Suppose, for example, that IRMQ is enabled and there are two MBs with the same ID, and FlexCAN starts receiving messages with that ID. Let us say that these MBs are the second and the fifth in the array. When the first message arrives, the matching algorithm finds the first match in MB number 2. The code of this MB is EMPTY, so the message is stored there. When the second message arrives, the matching algorithm finds MB number 2 again, but it is not "free-to-receive", so it keeps looking, finds MB number 5 and stores the message there. If yet another message with the same ID arrives, the matching algorithm finds out that there are no matching MBs that are "free-to-receive", so it decides to overwrite the last matched MB, which is number 5. In doing so, it sets the CODE field of the MB to indicate OVERRUN.

The ability to match the same ID in more than one MB can be exploited to implement a reception queue to allow more time for the CPU to service the MBs. By programming more than one MB with the same ID, received messages are queued into the MBs. The CPU can examine the Time Stamp field of the MBs to determine the order in which the messages arrived.

Matching to a range of IDs is possible by using ID Acceptance Masks. FlexCAN supports individual masking per MB. See the description of the Rx Individual Mask Registers (CAN_RXIMRx). During the matching algorithm, if a mask bit is asserted, then the corresponding ID bit is compared. If the mask bit is negated, the corresponding ID bit is a "don't care". Note that the Individual Mask Registers are implemented in RAM, so they are not initialized out of reset. Also, they can only be programmed while the module is in Freeze mode; otherwise, they are blocked by hardware.

FlexCAN also supports an alternate masking scheme with only four mask registers (CAN_RXMGMASK, CAN_RX14MASK and CAN_RX15MASK) for backwards compatibility with legacy applications. This alternate masking scheme is enabled when the IRMQ bit in the CAN_MCR Register is negated.

37.4.5 Move process

There are two types of move process: move-in and move-out.

37.4.5.1 Move-in

The move-in process is the copy of a message received by an Rx SMB to a Rx Mailbox that has matched it. Each Rx SMB has its own move-in process, but only one is performed at a given time as described ahead. The move-in starts only when the message held by the Rx SMB has a corresponding matching winner (see Matching process) and all of the following conditions are true:

- The CAN bus has reached or let past either:
  - The second bit of Intermission field next to the frame that carried the message that is in the Rx SMB
  - The first bit of an overload frame next to the frame that carried the message that is in the Rx SMB
- There is no ongoing matching process
- The destination Mailbox is not locked by the CPU
- There is no ongoing move-in process from another Rx SMB. If more than one move-in processes are to be started at the same time both are performed and the newest substitutes the oldest.

The term *pending move-in* is used throughout the documentation and stands for a move-to-be that still does not satisfy all of the aforementioned conditions.

The move-in is cancelled and the Rx SMB is able to receive another message if any of the following conditions is satisfied:

- The destination Mailbox is inactivated after the CAN bus has reached the first bit of Intermission field next to the frame that carried the message and its matching process has finished
- There is a previous pending move-in to the same destination Mailbox
- The Rx SMB is receiving a frame transmitted by the FlexCAN itself and the self-reception is disabled (CAN_MCR[SRXDIS] bit is asserted)
- Any CAN protocol error is detected

Note that the pending move-in is not cancelled if the module enters Freeze or Low-Power mode. It only stays on hold waiting for exiting Freeze and Low-Power mode and to be unlocked. If an MB is unlocked during Freeze mode, the move-in happens immediately.

The move-in process is the execution by the FlexCAN of the following steps:

1. Read all data words from the Rx SMB in accordance to the selected payload size for the Rx storage element.
2. Write all data words to the Rx Mailbox in accordance to the selected payload size for the Rx storage element. If the data size of the storage element is smaller than the original payload size described in the message's DLC field, the payload is truncated and the high order bytes that do not fit the destination size are lost.

3. Read the Control/Status and ID words from the Rx SMB.
4. Write Control/Status and ID words to the Rx Mailbox, and update the CODE field.

The move-in process is not atomic, in such a way that it is immediately cancelled by the inactivation of the destination Mailbox (see Mailbox inactivation) and in this case the Mailbox may be left partially updated, thus incoherent.

The BUSY Bit (least significant bit of the CODE field) of the destination Message Buffer is asserted while the move-in is being performed to alert the CPU that the Message Buffer content is temporarily incoherent.

37.4.5.2 Move-out

The move-out process is the copy of the content from a Tx Mailbox to the Tx SMB when a message for transmission is available (see Section "Arbitration process"). The move-out occurs in the following conditions:

- The first bit of Intermission field
- During Bus Off state when TX Error Counter is in the 124 to 128 range
- During Bus Idle state
- During Wait For Bus Idle state

The move-out process is not atomic. Only the CPU has priority to access the memory concurrently out of Bus Idle state. In Bus Idle, the move-out has the lowest priority to the concurrent memory accesses.

37.4.6 Data coherence

In order to maintain data coherency and FlexCAN proper operation, the CPU must obey the rules described in Transmit process and Receive process.

37.4.6.1 Transmission abort mechanism

The abort mechanism provides a safe way to request the abortion of a pending transmission. A feedback mechanism is provided to inform the CPU if the transmission was aborted or if the frame could not be aborted and was transmitted instead.

Two primary conditions must be fulfilled in order to abort a transmission:

- CAN_MCR[AEN] bit must be asserted
- The first CPU action must be the writing of abort code (0b1001) into the CODE field of the Control and Status word.

Active MBs configured for transmission must be aborted first before they can be updated. If the abort code is written to a Mailbox that is currently being transmitted or to a Mailbox that was already loaded into the Tx SMB for transmission, the write operation is blocked and the transmission is not disturbed. However, the abort request is captured and kept pending until one of the following conditions is satisfied:

- The module loses the bus arbitration
- There is an error during the transmission
- The module is put into Freeze mode
- The module enters the BusOff state
- There is an overload frame

If none of the conditions above are reached, the MB is transmitted correctly, the interrupt flag is set in the IFLAG register, and an interrupt to the CPU is generated (if enabled). The abort request is automatically cleared when the interrupt flag is set. On the other hand, if one of the above conditions is reached, the frame is not transmitted; therefore, the abort code is written into the CODE field, the interrupt flag is set in the IFLAG, and an interrupt is (optionally) generated to the CPU.

If the CPU writes the abort code before the transmission begins internally, then the write operation is not blocked; therefore, the MB is updated and the interrupt flag is set. In this way the CPU just needs to read the abort code to make sure the active MB was *safely inactivated*. Although the AEN bit is asserted and the CPU wrote the abort code, in this case the MB is inactivated and not aborted, because the transmission did not start yet. One Mailbox is only aborted when the abort request is captured and kept pending until one of the previous conditions are satisfied.

37.4.6.2 Mailbox inactivation

Inactivation is a mechanism provided to protect the Mailbox against updates by the FlexCAN internal processes, thus allowing the CPU to rely on Mailbox data coherence after having updated it, even in Normal mode.

Inactivation of transmission Mailboxes must be performed just when MCR[AEN] bit is deasserted.

If a Mailbox is inactivated, it participates in neither the arbitration process nor the matching process until it is reactivated. See Transmit process and Receive process for more detailed instructions on how to inactivate and reactivate a Mailbox.

To inactivate a Mailbox, the CPU must update its CODE field to INACTIVE (either 0b0000 or 0b1000).

Because the user is not able to synchronize the CODE field update with the FlexCAN internal processes, an inactivation can have the following consequences:

- A frame in the bus that matches the filtering of the inactivated Rx Mailbox may be lost without notice, even if there are other Mailboxes with the same filter
- A frame containing the message within the inactivated Tx Mailbox may be transmitted without setting the respective IFLAG

In order to perform a *safe inactivation* and avoid the above consequences for Tx Mailboxes, the CPU must use the Transmission Abort mechanism (see Transmission abort mechanism).

The inactivation automatically unlocks the Mailbox (see Mailbox lock mechanism).

37.4.6.3 Mailbox lock mechanism

Other than Mailbox inactivation, FlexCAN has another data coherence mechanism for the receive process. When the CPU reads the Control and Status word of an Rx MB with codes FULL or OVERRUN, FlexCAN assumes that the CPU wants to read the whole MB in an atomic operation, and therefore it sets an internal lock flag for that MB. The lock is released when the CPU reads the Free Running Timer (global unlock operation), or when it reads the Control and Status word of another MB regardless of its code. A CPU write into the C/S word also unlocks the MB, but this procedure is not recommended for normal unlock use because it cancels a pending-move and potentially may lose a received message. The MB locking prevents a new frame from being written into the MB while the CPU is reading it.

NOTE

The locking mechanism applies only to Rx MBs that have a code different than INACTIVE (0b0000) or EMPTY[1] (0b0100). Also, Tx MBs can not be locked.

Suppose, for example, that the second and the fifth MBs of the array are programmed with the same ID, and FlexCAN has already received and stored messages into these two MBs. Suppose now that the CPU decides to read MB number 5 and at the same time another message with the same ID is arriving. When the CPU reads the Control and Status word of MB number 5, this MB is locked. The new message arrives and the matching algorithm finds out that there are no "free-to-receive" MBs, so it decides to

1. In previous FlexCAN versions, reading the C/S word locked the MB even if it was EMPTY. This behavior is maintained when the IRMQ bit is negated.

override MB number 5. However, this MB is locked, so the new message can not be written there. It will remain in the Rx SMB waiting for the MB to be unlocked, and only then will be written to the MB.

If the MB is not unlocked in time and yet another new message with the same ID arrives, then the new message overwrites the one on the Rx SMB and there will be no indication of lost messages either in the CODE field of the MB or in the Error and Status Register.

While the message is being moved-in from the Rx SMB to the MB, the BUSY bit on the CODE field is asserted. If the CPU reads the Control and Status word and finds out that the BUSY bit is set, it should defer accessing the MB until the BUSY bit is negated.

Note

If the BUSY bit is asserted or if the MB is empty, then reading the Control and Status word does not lock the MB.

Inactivation takes precedence over locking. If the CPU inactivates a locked Rx MB, then its lock status is negated and the MB is marked as invalid for the current matching round. Any pending message on the Rx SMB will not be transferred anymore to the MB. An MB is unlocked when the CPU reads the Free Running Timer Register (see Section "Free Running Timer Register (CAN_TIMER)"), or the C/S word of another MB.

Lock and unlock mechanisms have the same functionality in both Normal and Freeze modes.

An unlock during Normal or Freeze mode results in the move-in of the pending message. However, the move-in is postponed if an unlock occurs during a low power mode (see Modes of operation), and it takes place only when the module resumes to Normal or Freeze modes.

37.4.7 Rx FIFO

The Rx FIFO is receive-only and is enabled by asserting the CAN_MCR[RFEN] bit. The reset value of this bit is zero to maintain software backward compatibility with previous versions of the module that did not have the FIFO feature.

CAUTION

Rx FIFO must not be enabled when CAN FD feature is enabled.

The FIFO is 6-message deep. The memory region occupied by the FIFO structure (both Message Buffers and FIFO engine) is described in Rx FIFO structure. The CPU can read the received messages sequentially, in the order they were received, by repeatedly reading a Message Buffer structure at the output of the FIFO.

The CAN_IFLAG1[BUF5I] (Frames available in Rx FIFO) is asserted when there is at least one frame available to be read from the FIFO. An interrupt is generated if it is enabled by the corresponding mask bit. Upon receiving the interrupt, the CPU can read the message (accessing the output of the FIFO as a Message Buffer) and the CAN_RXFIR register and then clear the interrupt. If there are more messages in the FIFO the act of clearing the interrupt updates the output of the FIFO with the next message and update the CAN_RXFIR with the attributes of that message, reissuing the interrupt to the CPU. Otherwise, the flag remains negated. The output of the FIFO is only valid whilst the CAN_IFLAG1[BUF5I] is asserted.

The CAN_IFLAG1[BUF6I] (Rx FIFO Warning) is asserted when the number of unread messages within the Rx FIFO is increased to 5 from 4 due to the reception of a new one, meaning that the Rx FIFO is almost full. The flag remains asserted until the CPU clears it.

The CAN_IFLAG1[BUF7I] (Rx FIFO Overflow) is asserted when an incoming message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox. The flag remains asserted until the CPU clears it.

Clearing one of those three flags does not affect the state of the other two.

An interrupt is generated if an IFLAG bit is asserted and the corresponding mask bit is asserted too.

A powerful filtering scheme is provided to accept only frames intended for the target application, reducing the interrupt servicing work load. The filtering criteria is specified by programming a table of up to 128 32-bit registers, according to CAN_CTRL2[RFFN] setting, that can be configured to one of the following formats (see also Rx FIFO structure):

- Format A: 128 IDAFs (extended or standard IDs including IDE and RTR)
- Format B: 256 IDAFs (standard IDs or extended 14-bit ID slices including IDE and RTR)
- Format C: 512 IDAFs (standard or extended 8-bit ID slices)

Note

A chosen format is applied to all entries of the filter table. It is not possible to mix formats within the table.

Every frame available in the FIFO has a corresponding IDHIT (Identifier Acceptance Filter Hit Indicator) that can read in the IDHIT field from C/S word, as shown in the Rx FIFO Structure description. Another way the CPU can obtain this information is by accessing the CAN_RXFIR register. The CAN_RXFIR[IDHIT] field refers to the message at the output of the FIFO and is valid while the CAN_IFLAG1[BUF5I] flag is asserted. The CAN_RXFIR register must be read only before clearing the flag, which guarantees that the information refers to the correct frame within the FIFO.

Up to 32 elements of the filter table are individually affected by the Individual Mask Registers (CAN_RXIMRx), according to the setting of CAN_CTRL2[RFFN], allowing very powerful filtering criteria to be defined. If the CAN_MCR[IRMQ] bit is negated, then the FIFO filter table is affected by CAN_RXFGMASK.

NOTE

For more information about the difference between FD and non-FD regarding this feature, see Table 37-1.

37.4.7.1 Rx FIFO under DMA Operation

The receive-only FIFO can support DMA, this feature is enabled by asserting both the CAN_MCR[RFEN] and CAN_MCR[DMA] bits. The reset value of CAN_MCR[DMA] bit is zero to maintain backward compatibility with previous versions of the module that did not have the DMA feature.

The DMA controller can read the received message by reading a Message Buffer structure at the FIFO output port at the 0x80-0x8C address range.

When CAN_MCR[DMA] is asserted the CPU must not access the FIFO output port address range. Before enabling the CAN_MCR[DMA], the CPU must service the IFLAGs asserted in the Rx FIFO region. Otherwise, these IFLAGs may show that the FIFO has data to be serviced, and mistakenly generate a DMA request. Before disabling the CAN_MCR[DMA], the CPU must perform a clear FIFO operation.

The CAN_IFLAG1[BUF5I] (Frames available in Rx FIFO) is asserted when there is at least one frame available to be read from the FIFO, consequently a DMA request is generated simultaneously. Upon receiving the request, the DMA controller can read the message (accessing the output of the FIFO as a Message Buffer). The DMA reading process must end by reading address 0x8C, which clears the CAN_IFLAG1[BUF5I] and updates both the FIFO output with the next message (if FIFO is not empty) and the

CAN_RXFIR register with the attributes of the new message. If there are more messages stored in the FIFO, the CAN_IFLAG1[BUF5I] will be re-asserted and another DMA request is issued. Otherwise, the flag remains negated.

The CAN_IFLAG1[BUF6I] and CAN_IFLAG1[BUF7I] are not used when the DMA feature is enabled.

When FlexCAN is working with DMA, the CPU does not receive any Rx FIFO interruption and must not clear the related IFLAGs. In addition, the related IMASKs are not used to mask the generation of DMA requests.

NOTE

For more information about the difference between FD and non-FD regarding this feature, see Table 37-1.

37.4.7.2 Clear FIFO Operation

When CAN_MCR[RFEN] is asserted, the clear FIFO operation is a feature used to empty FIFO contents. With CAN_MCR[RFEN] asserted the Clear FIFO occurs when the CPU writes 1 in CAN_IFLAG1[BUF0I]. This operation can only be performed in Freeze Mode and is blocked by hardware in other modes. This operation does not clear the FIFO IFLAGs, consequently the CPU must service all FIFO IFLAGs before execute the clear FIFO task.

When Rx FIFO is working with DMA, the clear FIFO operation clears the CAN_IFLAG1[BUF5I] and the DMA request is canceled.

CAUTION

Clear FIFO operation does not clear IFLAGs, except when CAN_MCR[DMA] is asserted, in this case only the CAN_IFLAG1[BUF5I] is cleared.

37.4.8 CAN protocol related features

This section describes the CAN protocol related features.

37.4.8.1 CAN FD ISO compliance

The CAN FD protocol has been improved to increase the failure detection capability that was in the original CAN FD protocol, which is also called non-ISO CAN FD by CAN in Automation (CiA). A three-bit stuff counter and a parity bit have been introduced in the

improved CAN FD protocol, now called ISO CAN FD. The CRC calculation has also been modified. All these improvements make the ISO CAN FD protocol incompatible with the non-FD CAN FD protocol. The non-ISO CAN FD is still supported by FlexCAN so that it can be used mainly during an intermediate phase, for evaluation and development purposes.

Therefore, it is strongly recommended to configure FlexCAN to the ISO CAN FD protocol by setting the ISOCANFDEN field in the CAN_CTRL2 register.

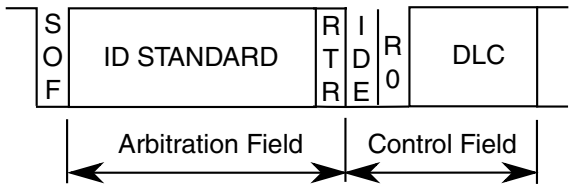
37.4.8.2 CAN FD frames

The ISO 11898-1 standard specifies the Classical Frame format compliant to ISO 11898-1 (2003) and introduces the CAN Flexible Data Rate Frame format. The Classical Frame format allows bit rates up to 1 Mbit/s and payloads up to 8 bytes per frame. The Flexible Data Rate Frame format allows bit rates higher than 1 Mbit/s and payloads longer than 8 bytes per frame. FlexCAN can receive and transmit CAN FD messages interleaved with Classical CAN messages.

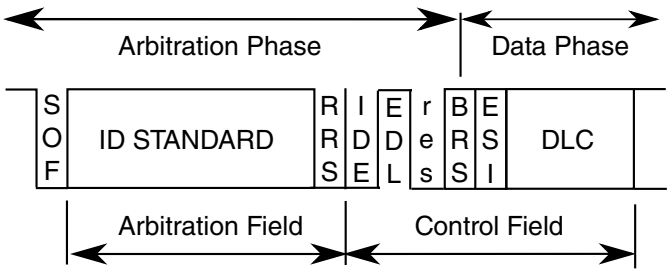
There are three additional control bits in the CAN FD frame. The Extended Data Length (EDL) bit enables a longer data payload with different data length coding. The Bit Rate Switch (BRS) bit decides whether the bit rate is switched inside a CAN FD format frame. The Error State Indicator (ESI) flag is transmitted dominant by error active nodes, and recessive by error passive nodes. There is no Remote Frames (see Remote frames) in the CAN FD format. A message configured to transmit a Remote Frame is always sent out in the Classical CAN format. When a FD frame is received and matches a mailbox, the RTR bit in the receiving message buffer is negated. The RTR bit must be considered in classical frames only.

CAN FD messages may be formatted as long frames where the data field exceeds 8 bytes, and may range from 12 up to 64 bytes. They can also be configured to support bit rate switching, where the control field, the data field, and the CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame. Messages in Classical CAN format are limited to transport a maximum payload of 8 bytes at nominal rate. The following figure illustrates the message formats for Classical and FD frames with either standard or extended ID.

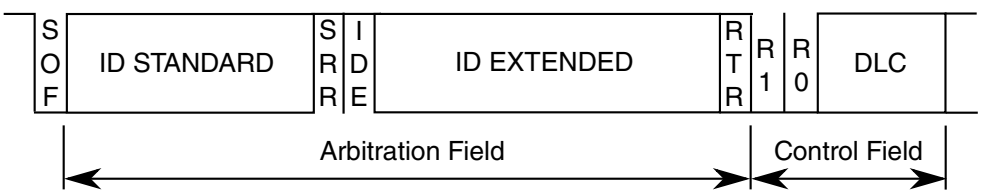
CAN Standard Format



CAN FD Standard Format



CAN Extended Format



CAN FD Extended Format

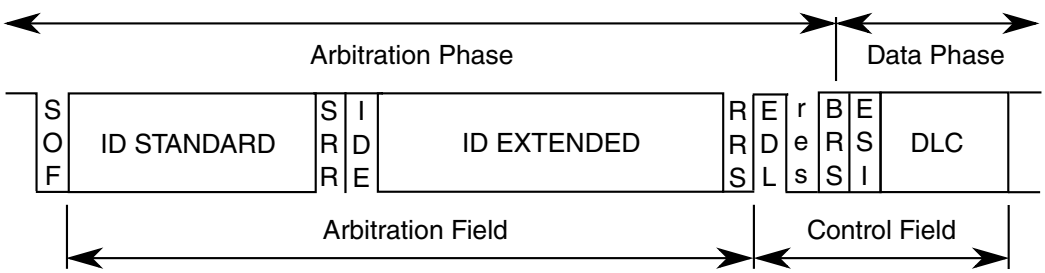


Figure 37-2. CAN message formats

The ability to receive and transmit CAN FD messages is enabled by the CAN_MCR[FDEN] bit. Either a recessive R0 bit in CAN frames with 11-bit identifiers or a recessive R1 bit in CAN frames with 29-bit identifiers are decoded as an EDL bit (not a reserved one). A CAN FD frame is recognized by a recessive EDL bit, while a

Classical CAN frame is recognized by a dominant EDL bit. The BRS bit specifies whether this frame switches the bit rate in its data phase. A long frame is decoded in accordance to the DLC field value (see DLC definition in Message buffer structure).

CAN FD messages can be transmitted with two different bit rates. The first part of a CAN FD frame, from the Start Of Frame (SOF) bit until the Bit Rate Switch (BRS) bit, also called the arbitration phase, is transmitted with the nominal bit rate based on a set of nominal CAN bit timing configuration values. The second part, from the BRS bit until the CRC Delimiter bit, also named the data phase, is transmitted with the data bit rate defined by a second set of CAN data bit timing configuration values. Finally, from the CRC Delimiter until the Intermission bits, the transmission resumes to nominal bit rate. In CAN FD frames with bit rate switching, the bit timing is changed inside the frame at the sample point of the BRS bit if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the CAN_CBT register (also by CAN_CTRL1 register for backward compatibility). Upon detecting a recessive BRS bit, the CAN data bit timing is used as defined by the CAN_FDCBT register.

NOTE

If the length of the time quantum in the nominal bit timing and the length of the time quantum in the data bit timing are not identical, a quantization error of up to one time quantum of the arbitration phase may be present as a phase error. This situation can occur after the switch from arbitration to data phase and will last until the next synchronization event. Thus, the length of the time quantum should be the same in nominal and data bit timing in order to minimize the chance of error frames on the CAN bus, and to optimize the clock tolerance in networks that use FD frames.

CAN_FDCTRL[FDRATE] enables the transmission of all frames with bit rate switching if the BRS bit in the selected Tx MB is set. If FDRATE is negated, the transmission is performed at nominal rate regardless of the BRS bit value. The CAN_FDCTRL[FDRATE] bit can be written any time but takes effect only for the next message transmitted or received.

The nominal bit timing is resumed at either the sample point of the CRC Delimiter bit or when an error is detected, whichever occurs first. The following figure describes the mechanism for entering and leaving the data phase when BRS bit is recessive.

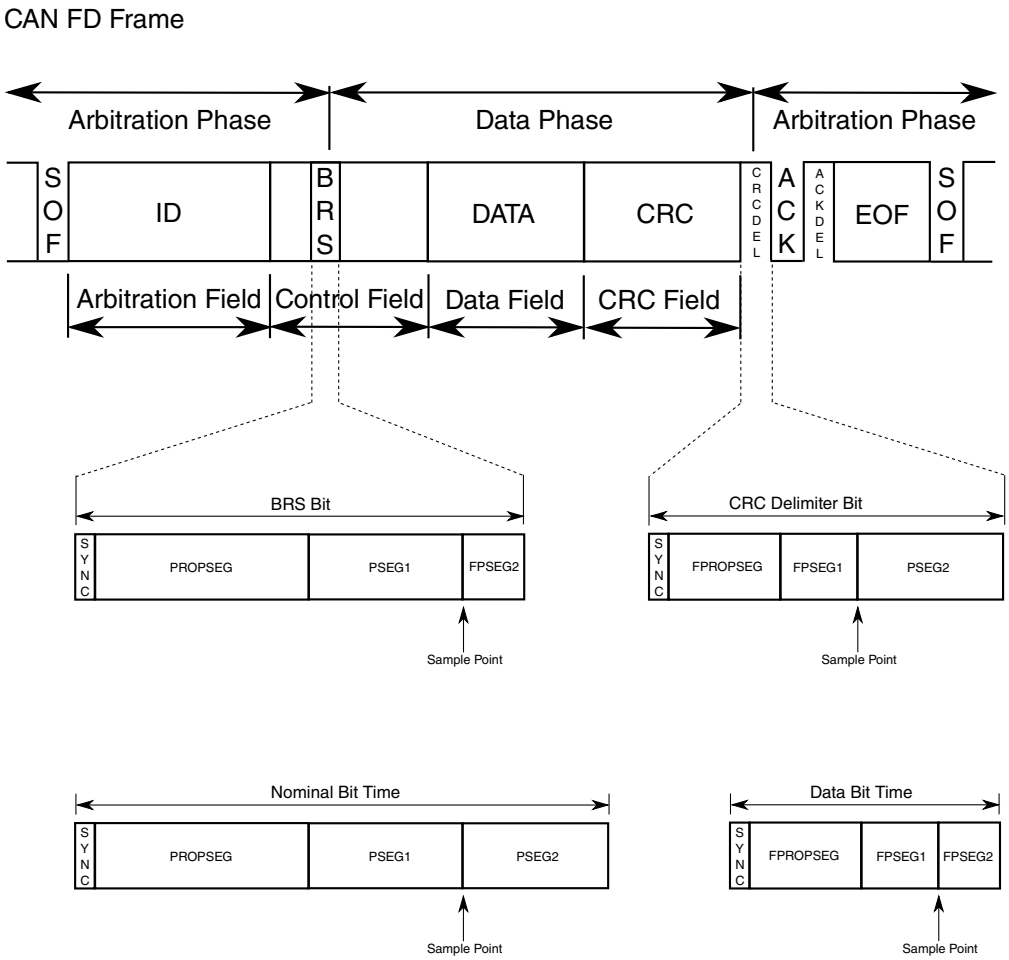


Figure 37-3. Bit rate switching mechanism for CAN FD messages

NOTE

In Classical CAN frames, the CRC delimiter is one single recessive bit. In CAN FD frames, the CRC delimiter may consist of one or two recessive bits. FlexCAN sends only one recessive bit as the CRC delimiter, but it accepts two recessive bits before the edge from recessive to dominant that starts the acknowledge slot. As a receiver, FlexCAN sends its acknowledge bit after the first CRC delimiter bit. In CAN FD frames, FlexCAN accepts a two-bit dominant ACK slot as a valid ACK to compensate for phase shifts between the receivers.

The maximum configurable bit rate in the CAN FD data phase depends on the clock frequency of CAN_PE sub-block. For example, with a CAN_PE clock frequency of 40MHz and the shortest configurable bit time of 5 time quanta, the bit rate in the data phase is 8 Mbit/s.[3]

The value of the ESI bit is determined either by the transmitter's error state at the start of the transmission, if the frame is originated in the FlexCAN node, or by the original transmitting node in case FlexCAN is acting as a gateway for the message. If the transmitter is error passive, ESI is transmitted recessive; otherwise, it is transmitted dominant. The permutations of the relationship between the written value and the transmitted value of the ESI are shown in this table.

Table 37-20. Written versus transmitted values of ESI field

FlexCAN fault confinement status at start of frame	ESI bit Of Tx MB	Transmitted ESI
Error active	0	0 (Error Active)
Error passive	0	1 (Error Passive)
Error active	1	1 (Error Passive)
Error passive	1	1 (Error Passive)

There are different CRC polynomials for different CAN frame formats. The first polynomial, CRC_15, is used for all frames in Classical CAN format. The second, CRC_17, is used for frames in CAN FD format with a data field up to sixteen bytes long. The third, CRC_21, is used for frames in CAN FD format with a data field longer than sixteen bytes. Each polynomial results in a Hamming Distance of 6. At the start of the frame, all three CRC polynomials are calculated concurrently. The CRC sequence to be transmitted is selected by the values of the EDL bit and the DLC bit field. When receiving a message, FlexCAN decodes EDL and DLC to select the adequate CRC polynomial to check for a CRC error.

In CAN FD format frames, stuff bits are included in the bit stream for CRC calculation. In Classical CAN format frames, stuff bits are not included. After the transmission of the last bit relevant to the CRC calculation, the CAN_FDCRC register stores the calculated CRC for the transmitted message, with the adequate length in accordance to the type of message, for both CAN FD and non-FD messages. The CAN_CRCR register reports a valid CRC for Classical CAN messages only.

In CAN FD format frames, the CAN bit stuffing method is changed for the CRC sequence so that the stuff bits are inserted at fixed positions. When FlexCAN is transmitting a CAN FD frame, a fixed stuff bit is inserted just before the first bit of the CRC sequence, even if the last bits of the preceding field do not fulfill the CAN stuff condition. Additional stuff bits are inserted after each fourth bit of the CRC sequence. The value of any fixed stuff bit is the inverse value of its preceding bit. When FlexCAN

3. The frequency used in this example may not be supported on this chip; it is shown only to demonstrate how the maximum configurable bit rate is calculated.

is receiving a CAN FD frame, it discards the fixed stuff bits from the bit stream for the CRC check. A Stuff Error is detected if the fixed stuff bit has the same value as its preceding bit.

FlexCAN detects errors in CAN FD frames the same way as in Classical CAN frames. The error counters RXERRCNT and TXERRCNT in the CAN_ECR register accumulate the counts of Rx and Tx errors, respectively, for both FD and non-FD frames indistinctly. There are two extra error counters (RXERRCNT_FAST and TXERRCNT_FAST) that accumulate Rx and Tx errors occurring in the data phase of CAN FD frames with the BRS bit set only. The rules for updating the error counters are the same for both CAN FD and non-FD frames (see CAN_ECR register).

Error Flags BITERR1, BITERR0, ACKERR, CRCERR, FRMERR and STFERR in the ESR1 register report errors in both CAN FD and non-FD frames. They also generate the ERRINT interrupt if CAN_CTRL1[ERRMSK] is asserted. The CAN_ESR1 register has additional error flags (BITERR1_FAST, BITERR0_FAST, CRCERR_FAST, FRMERR_FAST and STFERR_FAST) to individually indicate the occurrence of errors in the data phase of CAN FD frames with the BRS bit set. There is no ACKERR detected in the data phase of a CAN FD frame. Fault confinement status reported in CAN_ESR1[FLTCONF] is the same for both CAN FD and Classical CAN frames, and is based on RXERRCNT and TXERRCNT error counters only. Information contained in RXERRCNT_FAST and TXERRCNT_FAST counters may be considered as status to help detect the error nature related to the bit rate value.

When FlexCAN is in the data phase, either transmitting or receiving a CAN FD message, and detects an error, it immediately switches back to the arbitration phase and to the nominal rate to start an Error Flag.

Resynchronization and Hard Synchronization occur in CAN FD frames in the same way as in Classical CAN ones. Additionally, a Hard Synchronization is also performed at the recessive to dominant edge from EDL to R0 in CAN FD format frames. FlexCAN does not resynchronize while transmitting in the CAN FD data phase.

37.4.8.3 Transceiver Delay Compensation

The CAN FD protocol allows the transmission and reception of data at a higher bit rate than the nominal rate used in the arbitration phase when the message's BRS bit is set. This feature enables the use of rates up to 8 Mbps.

During the data phase of a CAN FD frame, the Transmitter detects a bit error if it cannot receive its own latest transmitted bit at the sample point of that bit. When bit rate switching is enabled (BRS bit is asserted), the length of the CAN bit time in the data

phase can become shorter than the transceiver's loop delay, thus impeding the correct comparison between the transmitted bit and the received bit within the current CAN bit time interval.

Note that the TDC process defines a secondary sample point where the transmitted bit is correctly compared with the received bit in order to check for bit errors.

The TDC mechanism can be enabled by the CAN_FDCTRL[TDCEN] bit and is effective only during the data phase of FD frames having the BRS bit set. It has no effect either on non-FD frames, or on FD frames transmitted at normal bit rate. The TDC is active from the sample point of the BRS bit until the sample point of the CRC Delimiter bit, provided the respective message under transmission has the BRS bit set. When it is active, a comparison is done between the real received bit and the delayed transmitted bit, where the delay is calculated based on the measured transceiver loop delay.

NOTE

The actual value of the CRC Delimiter bit is disregarded by transmitters using the Transceiver Delay Compensation mechanism. A global error at the end of the CRC Field will cause the receivers to send error frames that the transmitter will detect during Acknowledge or End of Frame.

For every transmitted FD frame having the BRS bit set, the delay measurement is triggered by the transition from the recessive EDL bit to the dominant R0 bit (as shown in the next figure). The loop delay is measured in Protocol Engine (PE) clock periods (CANCLK, see Protocol timing), from the transmitted EDL-R0 edge to the received EDL-R0 edge. The position of the secondary sample point is defined by the measured loop delay time added to an offset value specified in CAN_FDCTRL[TDCOFF]. CAN_FDCTRL[TDCVAL] bit field stores the result of this calculation. The TDCVAL value saturates at its maximum value of 63 CANCLK when the delay measurement is too long.

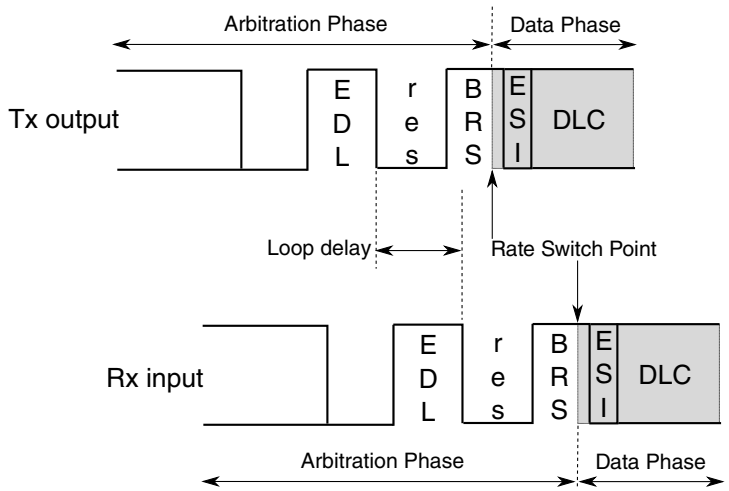


Figure 37-4. Transceiver loop delay measurement

The measured loop delay is not enough to be used to define the secondary sample point because it relates to the CAN bit edges. The transceiver delay compensation offset TDCOFF is used to shift the secondary sample point from the edge to an intermediate point inside the bit time, far away from its edges. Therefore, the TDCOFF value cannot be larger than the CAN bit duration in the data phase.

If the secondary sample point is set very near the CAN bit edge (SYNC field), then problems may occur during the bit sampling in the data phase. For the TDC to work reliably, the offset has to use optimal settings. To be sure the bit sampling is performed in the best region, the TDC offset should be configured as shown in this equation:

$$\text{Offset} = \text{PSEG1} + \text{PROPSEG} + 2$$

The following figure shows the SSP position when these settings are used.

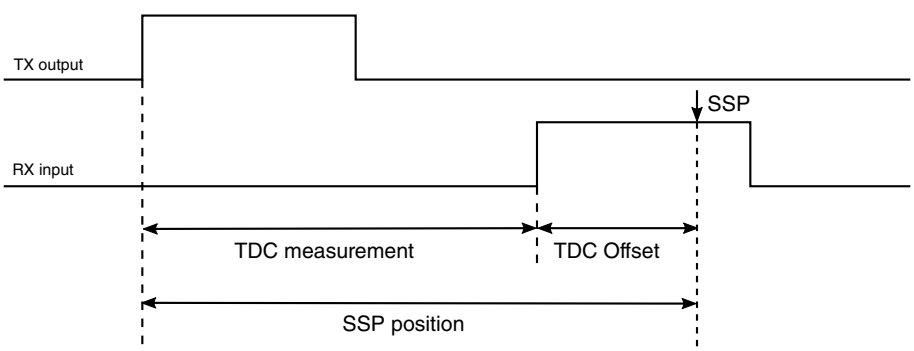


Figure 37-5. SSP position with optimal values

During the data phase of CAN FD frames with bit rate switching enabled, at the onset of every Tx CAN bit, the transmitted Tx bit value is temporarily stored in a buffer and a time countdown based on TDCVAL is started which ends with the comparison of the

received Rx bit (delayed by the external loop delay plus the specified offset) with the stored Tx bit. If a bit error is detected at the secondary sample point, the FlexCAN issues an error flag to the CAN bus at the next sample point.

During the arbitration phase the delay compensation is always disabled. The maximum delay which can be compensated by the FlexCAN's transceiver delay compensation during the data phase is 3 CAN bit times - 2 Tq. Beyond this limit, the CAN_FDCTRL[TDCFAIL] flag is set to indicate when the Transceiver Delay Compensation mechanism is out of range, unable to compensate the transceiver loop delay.

37.4.8.4 Remote frames

Remote frame is a special kind of frame. The user can program a mailbox to be a Remote Request Frame by configuring the mailbox as Transmit with the RTR bit set to '1'. After the remote request frame is transmitted successfully, the mailbox becomes a Receive Message Buffer, with the same ID as before.

When a remote request frame is received by FlexCAN, it can be treated in three ways, depending on Remote Request Storing (CTRL2[RRS]) and Rx FIFO Enable (MCR[RFEN]) bits:

- If RRS is negated the frame's ID is compared to the IDs of the Transmit Message Buffers with the CODE field 0b1010. If there is a matching ID, then this mailbox frame will be transmitted. Note that if the matching mailbox has the RTR bit set, then FlexCAN will transmit a remote frame as a response. The received remote request frame is not stored in a receive buffer. It is only used to trigger a transmission of a frame in response. The mask registers are not used in remote frame matching, and all ID bits (except RTR) of the incoming received frame should match. In the case that a remote request frame is received and matches a mailbox, this message buffer immediately enters the internal arbitration process, but is considered as a normal Tx mailbox, with no higher priority. The data length of this frame is independent of the DLC field in the remote frame that initiated its transmission.
- If RRS is asserted the frame's ID is compared to the IDs of the receive mailboxes with the CODE field 0b0100, 0b0010 or 0b0110. If there is a matching ID, then this mailbox will store the remote frame in the same fashion of a data frame. No automatic remote response frame will be generated. The mask registers are used in the matching process.
- If RFEN is asserted FlexCAN will not generate an automatic response for remote request frames that match the FIFO filtering criteria. If the remote frame matches one of the target IDs, it will be stored in the FIFO and presented to the CPU. Note that for

filtering formats A and B, it is possible to select whether remote frames are accepted or not. For format C, remote frames are always accepted (if they match the ID). Remote Request Frames are considered as normal frames, and generate a FIFO overflow when a successful reception occurs and the FIFO is already full.

NOTE

There is no remote frame in the CAN FD format. The RTR bit is replaced by a fixed dominant RRS bit. FlexCAN receives and transmits remote frames in the Classical CAN format.

37.4.8.5 Overload frames

FlexCAN does transmit overload frames due to detection of following conditions on CAN bus:

- Detection of a dominant bit in the first/second bit of Intermission
- Detection of a dominant bit at the 7th bit (last) of End of Frame field (Rx frames)
- Detection of a dominant bit at the 8th bit (last) of Error Frame Delimiter or Overload Frame Delimiter

37.4.8.6 Time stamp

The value of the Free Running Timer is sampled at the beginning of the Identifier field on the CAN bus, and is stored at the end of "move-in" in the TIME STAMP field, providing network behavior with respect to time.

When the TIMER_SRC bit in CAN_CTRL2 register is asserted, the Free Running Timer is continuously clocked by an external time tick.

When the TIMER_SRC bit in CAN_CTRL2 register is negated, the Free Running Timer is clocked by the FlexCAN bit-clock, which defines the baud rate on the CAN bus. During a message transmission/reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate.

The Free Running Timer is not incremented during Disable, Doze, Stop, and Freeze modes. It can be reset upon a specific frame reception, enabling network time synchronization. See the TSYN description in Control 1 Register (CAN_CTRL1).

37.4.8.7 Protocol timing

The following figure shows the structure of the clock generation circuitry that feeds the CAN Protocol Engine (PE) submodule. The clock source bit CLKSRC in the CAN_CTRL1 Register defines whether the internal clock is connected to the output of a crystal oscillator (Oscillator Clock) or to the Peripheral Clock. In order to guarantee reliable operation, the clock source should be selected while the module is in Disable Mode (MDIS bit set in the Module Configuration Register).

NOTE

Refer to the clock distribution chapter (module clocks table) to identify the proper clock source.

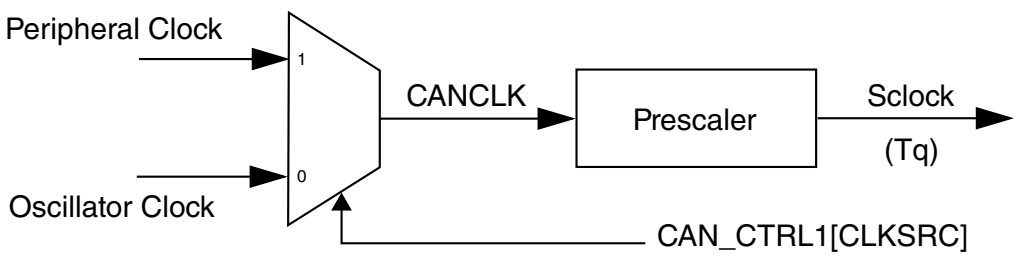


Figure 37-6. CAN engine clocking scheme

The oscillator clock should be selected whenever a tight tolerance (up to 0.1%) is required in the CAN bus timing. The crystal oscillator clock has better jitter performance than the peripheral clock.

The FlexCAN module supports a variety of means to setup bit timing parameters that are required by the CAN protocol. The Control 1 Register (CAN_CTRL1) has various fields used to control bit timing parameters: PRESDIV, PROPSEG, PSEG1, PSEG2 and RJW.

The CAN Bit Timing register (CAN_CBT) extends the range of the CAN bit timing variables in CAN_CTRL1. The CAN FD Bit Timing register (CAN_FDCBT) provides a second set of CAN bit timing variables to be applied at the data phase of CAN FD frames with the Bit Rate Switch (BRS) set.

NOTE

When the CAN FD feature is enabled, always set CAN_CBT[BTF] and configure the CAN bit timing variables in CAN_CBT. See CAN Bit Timing Register (CBT).

The PRESDIV field (as well as its extended range EPRESDIV and FDPRESDIV for the data phase bits of CAN FD messages) defines the Prescaler Value (see the equation below) that generates the Serial Clock (Sclock), whose period defines the 'time quantum' used to compose the CAN waveform. A time quantum (Tq) is the atomic unit of time handled by the CAN engine.

$$Tq = \frac{(PRESDIV + 1)}{f_{CANCLK}}$$

The bit rate, which defines the rate the CAN message is either received or transmitted, is given by the formula:

CAN Bit Time = (Number of Time Quanta in 1 bit time) * Tq

$$\text{Bit Rate} = \frac{1}{\text{CAN Bit Time}}$$

A bit time is subdivided into three segments[1] (see Figure 37-7, Figure 37-8 and Table 37-21):

- SYNC_SEG: This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section
- Time Segment 1: This segment includes the Propagation Segment and the Phase Segment 1 of the CAN standard. It can be programmed by setting the PROPSEG and the PSEG1 fields of the CAN_CTRL1 Register so that their sum (plus 2) is in the range of 2 to 16 time quanta. When CAN_CBT[BTF] bit is asserted, FlexCAN uses EPROPSEG and EPSEG1 fields from CAN_CBT register so that their sum (plus 2) is in the range of 2 to 96 time quanta. For messages in CAN FD format with the BRS bit set, FlexCAN uses FDPROPSEG and FDPSEG1 from CAN_FDCBT instead, so that their sum (plus 1) is in the range of 2 to 39 time quanta.
- Time Segment 2: This segment represents the Phase Segment 2 of the CAN standard. It can be programmed by setting the PSEG2 field of the CAN_CTRL1 Register (plus 1) to be 2 to 8 time quanta long. When CAN_CBT[BTF] bit is asserted, FlexCAN uses EPSEG2 fields of CAN_CBT register so that its value (plus 1) is in the range of 2 to 32 time quanta. For messages in CAN FD format with the BRS bit set, FlexCAN

1. For further explanation of the underlying concepts, see ISO 11898-1. See also the CAN 2.0A/B protocol specification for bit timing.

uses FDPSEG2 from CAN_FDCBT instead, so that its value (plus 1) is in the range of 2 to 8 time quanta. The Time Segnment 2 cannot be smaller than the Information Processing Time (IPT), which value is 2 time quanta in FlexCAN.

NOTE

The bit time defined by the above time segments must not be smaller than 5 time quanta. For bit time calculations, use an Information Processing Time (IPT) of 2, which is the value implemented in the FlexCAN module.

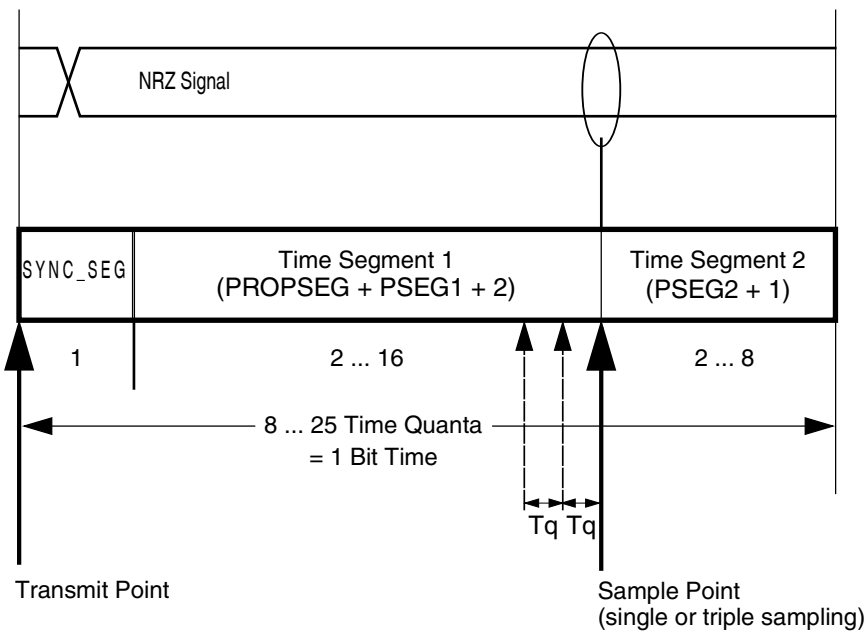


Figure 37-7. Segments within the bit time (example using CAN_CTRL1 bit timing variables for Classical CAN format)

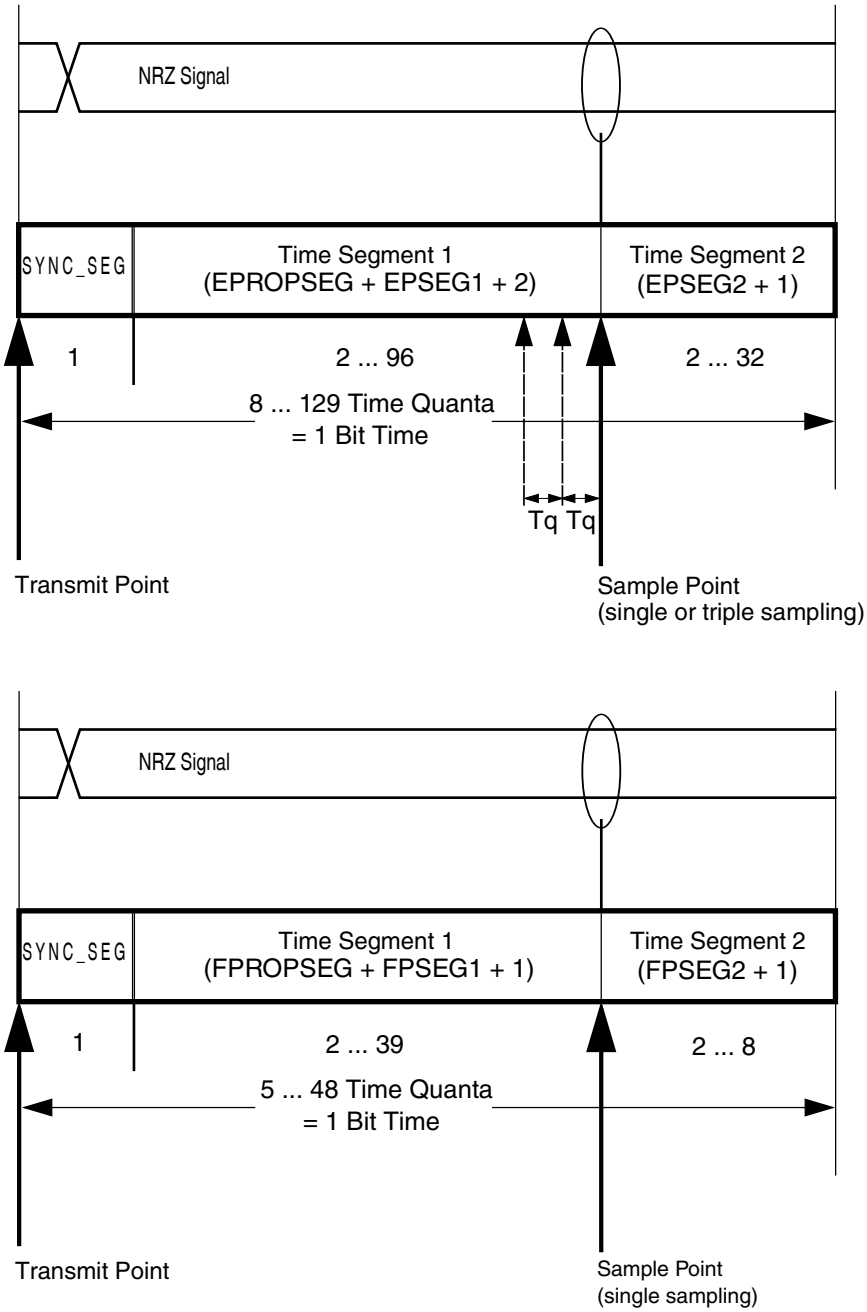


Figure 37-8. Segments within the bit time (example using CAN_CBT and CAN_FDCBT bit timing variables for CAN FD format)

Table 37-21. Time segment syntax

Syntax	Description
SYNC_SEG	System expects transitions to occur on the bus during this period.
TSEG1	Corresponds to the sum of PROPSEG and PSEG1.
TSEG2	Corresponds to the PSEG2 value.

Table continues on the next page...

Table 37-21. Time segment syntax (continued)

Syntax	Description
Transmit Point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample Point	A node samples the bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

The following table gives some examples of the CAN compliant segment settings for Classical CAN format (Bosch CAN 2.0B) (non-FD) messages.

Table 37-22. Bosch CAN 2.0B standard compliant bit time segment settings

Time segment 1	Time segment 2	Re-synchronization jump width
5 .. 10	2	1 .. 2
4 .. 11	3	1 .. 3
5 .. 12	4	1 .. 4
6 .. 13	5	1 .. 4
7 .. 14	6	1 .. 4
8 .. 15	7	1 .. 4
9 .. 16	8	1 .. 4

Note

The user must ensure the bit time settings are in compliance with the CAN Protocol standard (ISO 11898-1).

Whenever CAN bit is used as a measure of time duration (e.g. estimating the occurrence of a CAN bit event in a message), the number of peripheral clocks in one CAN bit (NumClkBit) can be calculated as:

$$\text{NumClkBit} = \frac{f_{SYS}}{f_{CANCLK}} \text{ X } (\text{PRESDIV} + 1) \text{ X } (\text{PROPSEG} + \text{PSEG1} + \text{PSEG2} + 4)$$

where:

- NumClkBit is the number of peripheral clocks in one CAN bit;
- f_{CANCLK} is the Protocol Engine (PE) Clock (see Figure "CAN Engine Clocking Scheme"), in Hz;
- f_{SYS} is the frequency of operation of the system (CHI) clock, in Hz;
- PSEG1 is the value in CAN_CTRL1[PSEG1] field;
- PSEG2 is the value in CAN_CTRL1[PSEG2] field;
- PROPSEG is the value in CAN_CTRL1[PROPSEG] field;
- PRESDIV is the value in CAN_CTRL1[PRESDIV] field.

The formula above is also applicable to the alternative CAN bit timing variables described in the CAN Bit Timing Register (CAN_CBT) and also to the CAN FD Bit Timing Register (CAN_FDCBT).

For example, 180 CAN bits = (180 x NumClkBit) peripheral clock periods.

37.4.8.8 Arbitration and matching timing

During normal reception and transmission, the matching, arbitration, move-in and move-out processes are executed during certain time windows inside the CAN frame, as shown in the following figures.

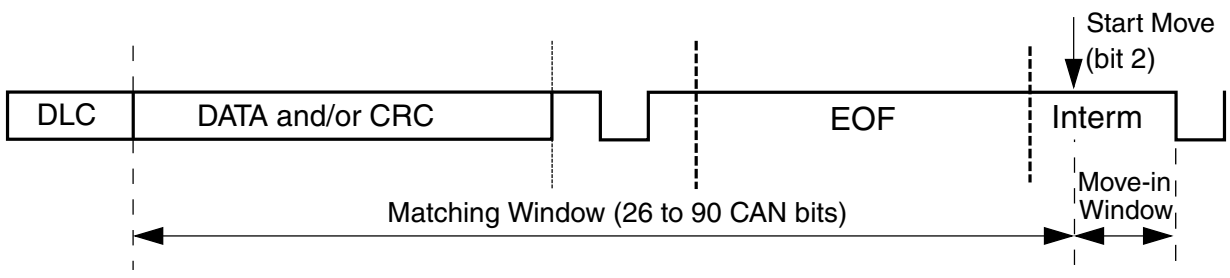


Figure 37-9. Matching and move-in time windows

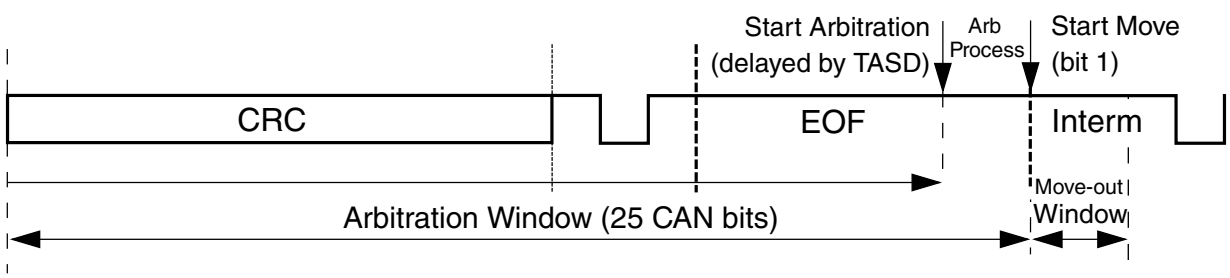


Figure 37-10. Arbitration and move-out time windows

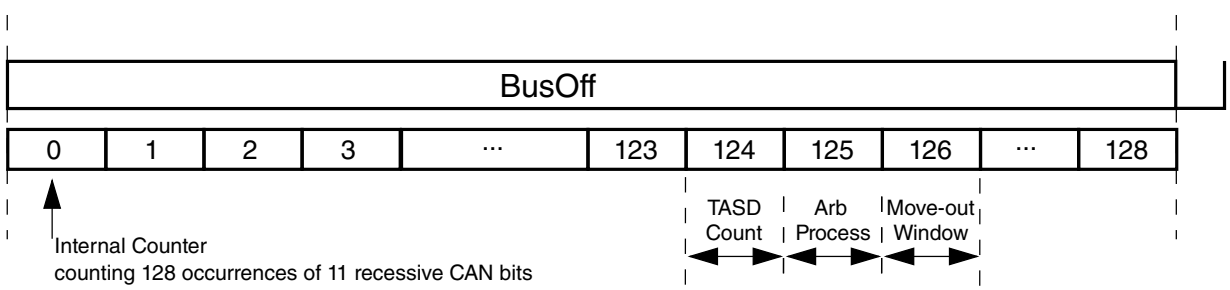


Figure 37-11. Arbitration at the end of bus off and move-out time windows

NOTE

In the preceding figures, the matching and arbitration timing does not take into account the delay caused by the concurrent

memory access due to the CPU or other internal FlexCAN sub-blocks.

37.4.8.9 Tx Arbitration start delay

The Tx Arbitration Start Delay (TASD) bit field in Control 2 register (CAN_CTRL2[TASD]) is a variable that indicates the number of CAN bits used by FlexCAN to delay the Tx Arbitration process start point from the first bit of CRC field of the current frame. This variable can be written only in Freeze mode because it is blocked by hardware in other modes.

The transmission performance is impacted by the ability of the CPU to reconfigure Message Buffers (MBs) for transmission after the end of the internal Arbitration process, where FlexCAN finds the winner MB for transmission (see Arbitration process). If the Arbitration ends too early before the first bit of Intermission field, then there is a chance that the CPU reconfigures some Tx MBs and the winner MB is no longer the best candidate to be transmitted.

TASD is useful to optimize the transmission performance by defining the Arbitration start point, as shown in the next figure, based on factors such as:

- The peripheral-to-oscillator clock ratio
- CAN bit timing variables that determine the CAN bit rate
- The number of Message Buffers (MBs) in use by the Matching and Arbitration processes.

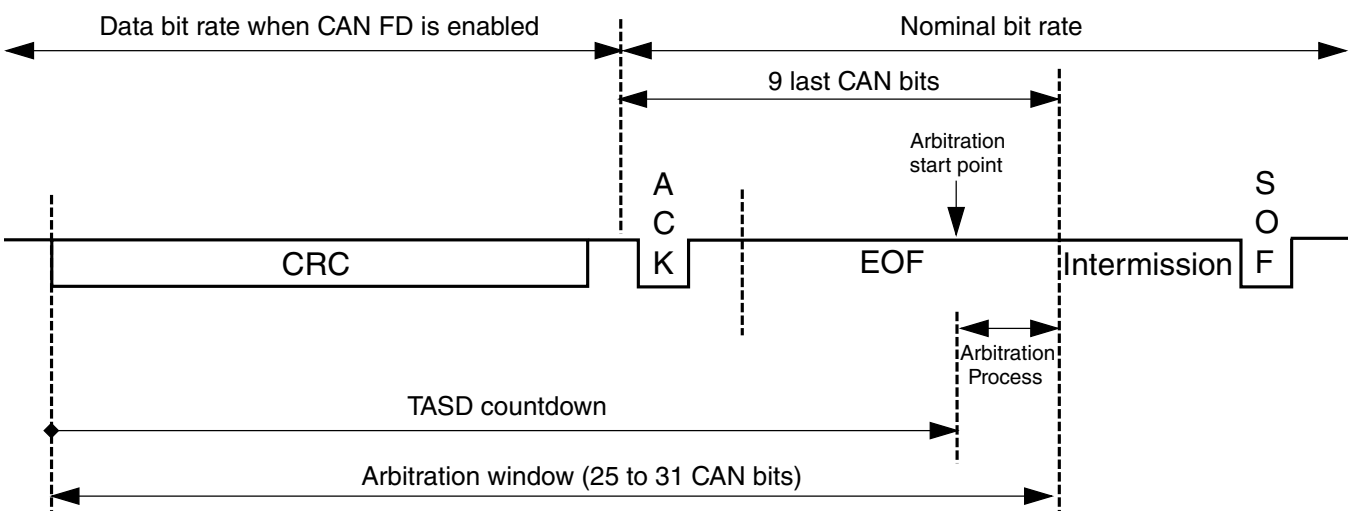


Figure 37-12. Optimal Tx Arbitration start point

The duration of an Arbitration process, in terms of CAN bits, is directly proportional to the number of available MBs and to the CAN bit rate, and inversely proportional to the peripheral clock frequency.

The optimal Arbitration timing is that in which the last MB is scanned right before the first bit of the Intermission field of a CAN frame. For instance, if there are few MBs and the peripheral/oscillator clock ratio is high and the CAN baud rate is low, then the Arbitration can be placed closer to the frame's end, adding more delay to its start point, and vice-versa.

If TASD is set to 0 then the Arbitration start is not delayed and more time is reserved for Arbitration. On the other hand, if TASD is close to 24 then the CPU can configure a Tx MB later and less time is reserved for Arbitration. If too little time is reserved for Arbitration the FlexCAN may be not be able to find a winner MB in time to be transmitted with the best chance to win the bus arbitration against external nodes on the CAN bus.

The optimal TASD value can be calculated as follows:

For CAN FD frames and $(MAXMB + 1) \leqslant NMB_{END}$

$$TASD = 31 - \frac{2 * (MAXMB + 1) + 4}{CPCB_N}$$

For CAN FD frames and $(MAXMB + 1) > NMB_{END}$

$$TASD = 22 - \frac{2 * (MAXMB + 1) - NMB_{END}}{CPCB_F}$$

For non-FD frames

$$TASD = 25 - \frac{2 * (MAXMB + 1) + 4}{CPCB}$$

where:

$$NMB_{END} = \frac{(9 * CPCB_N) - 4}{2}$$

$$BITRATE_N = \left(\frac{f_{CANCLK}}{[1 + (EPSEG1 + 1) + (EPSEG2 + 1) + (EPROPSEG + 1)] \times (EPRESDIV + 1)} \right)$$

$$BITRATE_F = \left(\frac{f_{CANCLK}}{[1 + (FPSEG1 + 1) + (FPSEG2 + 1) + FPROPSEG] \times (FPRESDIV + 1)} \right)$$

$$CPCB_N = \frac{f_{SYS}}{BITRATE_N}$$

$$CPCB_F = \frac{f_{SYS}}{BITRATE_F}$$

$$CPCB = CPCB_N$$

- MAXMB is the value in CAN_CTRL1[MAXMB] field
- NMB_{END} is the number of Message Buffers that can be scanned by the Arbitration process during the 9 last CAN bits at the end of a frame, see the figure above
- $BITRATE_N$ is the CAN bit rate in bits per second calculated by the nominal CAN bit time variables
- $BITRATE_F$ is the CAN bit rate in bits per second calculated by the data CAN bit time variables
- $CPCB_N$ is the number of peripheral clocks per CAN bit in nominal bit rate for CAN FD frames
- $CPCB_F$ is the number of peripheral clocks per CAN bit in data bit rate for CAN FD frames
- CPCB is the number of peripheral clocks per CAN bit for non-FD frames
- f_{CANCLK} is the oscillator clock, in Hz
- f_{SYS} is the peripheral clock, in Hz
- EPSEG1 is the value in CAN_CBT[EPSEG1] field (CAN_CTRL1[PSEG1] can also be used)
- EPSEG2 is the value in CAN_CBT[EPSEG2] field (CAN_CTRL1[PSEG2] can also be used)
- EPROPSEG is the value in CAN_CBT[EPROPSEG] field (CAN_CTRL1[PROPSEG] can also be used)

- EPRESDIV is the value in CAN_CBT[EPRESDIV] field (CAN_CTRL1[PRESDIV] can also be used)
- FPSEG1 is the value in CAN_FDCBT[FPSEG1] field
- FPSEG2 is the value in CAN_FDCBT[FPSEG2] field
- FPROPSEG is the value in CAN_FDCBT[FPROPSEG] field
- FPRESDIV is the value in CAN_FDCBT[FPRESDIV] field

See also Protocol timing for more details.

The following tables give the TASD value calculated for some configuration cases.

Case 1:

- Clock ratio = 2:1 (example: peripheral clock 80 MHz and oscillator clock 40 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 37-23. TASD values:

Number of Message Buffers	TASD value	Maximum Bit Rate in Data Phase (Mbaud)
16	24	Invalid
32	24	8.0

Case 2:

- Clock ratio = 1:1 (example: peripheral clock 40 MHz and oscillator clock 40 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 37-24. TASD values:

Number of Message Buffers	TASD value	Maximum Bit Rate in Data Phase (Mbaud)
16	24	Invalid
32	23	6.67

Case 3:

- Clock ratio = 2:1 (example: peripheral clock 40 MHz and oscillator clock 20 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 37-25. TASD values:

Number of Message Buffers	TASD value	Maximum Bit Rate in Data Phase (Mbaud)
16	24	Invalid
32	23	4.0

37.4.9 Clock domains and restrictions

The FlexCAN module has two clock domains asynchronous to each other:

- The Bus Domain feeds the Control Host Interface (CHI) submodule and is derived from the peripheral clock.
- The Oscillator Domain feeds the CAN Protocol Engine (PE) submodule and is derived directly from a crystal oscillator clock, so that very low jitter performance can be achieved on the CAN bus.

When CAN_CTRL1[CLKSRC] bit is set, synchronous operation occurs because both domains are connected to the peripheral clock (creating a 1:1 ratio between the peripheral and oscillator clocks).

When the two domains are connected to clocks with different frequencies and/or phases, there are restrictions on the frequency relationship between the two clock domains. In the case of asynchronous operation, the Bus Domain clock frequency must always be greater than the Oscillator Domain clock frequency.

NOTE

Asynchronous operation with a 1:1 ratio between peripheral and oscillator clocks is not allowed.

When doing matching and arbitration, FlexCAN needs to scan the whole Message Buffer memory during the time slot of one CAN frame, comprised of a number of CAN bits. In order to have sufficient time to do that, the following requirements must be observed:

- The peripheral clock frequency can not be smaller than the oscillator clock frequency
- There must be a minimum number of peripheral clocks per CAN bit, as specified in the table shown below

Table 37-26. Minimum number of peripheral clocks per CAN bit for Classical CAN format

Number of Mailboxes	Value of CAN_MCR[RFEN]	Minimum number of peripheral clocks per CAN bit
16	0	16
32	0	16
16	1	16
32	1	17

For classical frame format, the minimum number of peripheral clocks per CAN bit specified in the preceding table determines the minimum peripheral clock frequency for a given number of Mailboxes and for an expected CAN bit rate. The CAN bit rate depends

on the number of time quanta in a CAN bit, that can be defined by adjusting one or more of the bit timing values contained in either the Control 1 Register (CAN_CTRL1) or CAN Bit Time register (CAN_CBT). The time quantum (Tq) is defined in Protocol timing. The minimum number of time quanta per CAN bit must be 8; therefore, the oscillator clock frequency should be at least 8 times the CAN bit rate.

For CAN FD frame format, there are some constraints that need to be satisfied. The number of peripheral clocks per CAN bit in nominal bit rate (NumClkNomBit) can be calculated by the equation below.

$$\text{NumClkNomBit} = \frac{f_{SYS}}{f_{CANCLK}} \text{ X } (\text{PRESDIV} + 1) \text{ X } (\text{PROPSEG} + \text{PSEG1} + \text{PSEG2} + 4)$$

$$= \frac{f_{SYS}}{\text{NomBitRate}}$$

where PRESDIV, PSEG1 and PSEG2 are CAN bit time values in CTRL1 register. Alternatively, EPRESDIV, EPSEG1 and EPSEG2 values in CBT register can be used instead. NumClkNomBit can also be calculated as a function of the expected nominal bit rate used in the Arbitration Phase (NomBitRate) as shown in the equation above.

The number of CAN bits in the Data Phase of a FD Frames with the BRS bit set (fast CAN bits, in short) depends on the number of data bytes in the payload. The number of fast CAN bits (NumOfFastBits) can be determined in the table below. The less the number of data bytes, the less the number of fast CAN bits, and less time is available for FlexCAN to scan the whole Message Buffer memory during the internal matching and arbitration processes.

Table 37-27. Number of fast CAN bits in a CAN FD frame

Minimum number of data bytes	DLC field	NumOfFastBits
0	0x0	21
1	0x1	29
2	0x2	37
3	0x3	45
4	0x4	53
5	0x5	61
6	0x6	69
7	0x7	77
8	0x8	85
12	0x9	117
16	0xA	149

Table continues on the next page...

Table 37-27. Number of fast CAN bits in a CAN FD frame (continued)

Minimum number of data bytes	DLC field	NumOfFastBits
20	0xB	186
24	0xC	218
32	0xD	282
48	0xE	410
64	0xF	538

The critical part of a CAN FD frame is during the Data Phase, where the CAN bit rate is faster than in the Arbitration Phase. The minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) can be calculated to guarantee that enough time is available for FlexCAN to scan the Message Buffer memory during reception and transmission. The equation below calculates this constraint.

$$MinNumClkFastBit_A = \frac{(8.5 \times MaxNumOfMb) + 64 - (9 \times NumClkNomBit)}{NumOfFastBits}$$

where MaxNumOfMb is the maximum number of available Mailboxes defined in CAN_MCR[MAXMB].

The clock domain crossing circuit between the CHI and PE sub-blocks also imposes a minimum number of peripheral clocks per fast CAN bit for the handshake mechanism to work properly without loosing status information through the interface, as shown in the equation below.

$$MinNumClkFastBit_B = 3 \times \left(1 + \frac{f_{SYS}}{f_{CANCLK}} \right)$$

Therefore, the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) is determined by the larger of the two values calculated above.

$$MinNumClkFastBit = Maximum\ (MinNumClkFastBit_A, MinNumClkFastBit_B)$$

Then, the maximum CAN bit rate in the Data Phase of CAN FD frames (DataBitRateMAX) can be calculated as below.

$$DataBitRate_{MAX} = \frac{f_{CANCLK}}{ROUNDUP\left(\frac{MinNumClkFastBit \times f_{CANCLK}}{f_{SYS}}\right)}$$

The peripheral and oscillator clock frequencies, the maximum number of mailboxes and the expected nominal bit rate affect the maximum data bit rate attainable by FlexCAN in CAN FD mode. Besides, the data bit rate depends on the minimum payload size of FD frames used in a given application.

To illustrate how the CAN FD bit rate is affected by the configuration of FlexCAN variables, an application example with the peripheral and oscillator clock frequencies set to 50 MHz and 40 MHz, respectively, is considered.

Step 1 - Considering the nominal bit rate as 1 Mbps, the number of peripheral clocks per CAN bit in nominal bit rate is calculated as below.

$$NumClkNomBit = \frac{50 \times 10^6}{1 \times 10^6} = 50$$

Step 2 - The number of fast CAN bits (NumOfFastBits) is determined in the table presented above. For example, if the minimum payload in FD frames is 8 bytes, then there are 85 CAN bits in the Data Phase.

Step 3 - Assuming the maximum number of mailboxes is 96, the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) can be calculated.

$$MinNumClkFastBit_A = \frac{(8.5 \times 96) + 64 - (9 \times 50)}{85} = 5.06$$

$$MinNumClkFastBit_B = 3 \times \left(1 + \frac{50}{40}\right) = 6.75$$

$$MinNumClkFastBit = Maximum\ (\ 5.06, 6.75\ ) = 6.75$$

Step 4 - The maximum CAN bit rate in the Data Phase can be finally found.

$$\text{DataBitRate}_{MAX} = \frac{40 \times 10^6}{\text{ROUNDUP}\left(\frac{6.75 \times 40 \times 10^6}{50 \times 10^6}\right)} = 6.667 \text{ Mbps}$$

As demonstrated in this example, even though the oscillator clock frequency (40 MHz) is adequate to generate a data rate of 8 Mbps in CAN FD mode, the specific FlexCAN configuration limits this rate to 6.667 Mbps. This limitation is mainly due to the low peripheral clock frequency that imposes the MinNumClkFastBitB bound.

The table below shows the maximum data rate for CAN FD according to clock frequencies, payload size and number of available mailboxes. See in this table that, for some cases, if the number of available mailboxes is reduced, the FlexCAN can then achieve a data rate up to 8 Mbps.

Table 37-28. Maximum CAN bit rate in Data Phase on CAN FD frames

Peripheral clock frequency (MHz)	Payload size	Number of available mailboxes	Maximum data rate (Mbps)
40	8	94	6.667
40	8	114	5.0
40	12	117	6.667
40	12	128	5.714
50	12 to 64	128	6.667
60	8	126	8.0
60	12	128	8.0
67	6	128	8.0
80	3	128	8.0
100	0	128	8.0

37.4.10 Modes of operation details

The FlexCAN module has functional modes and low-power modes. See Modes of operation for an introductory description of all the modes of operation. The following sub-sections contain functional details on Freeze mode and the low-power modes.

CAUTION

"Permanent Dominant" failure on CAN Bus line is not supported by FlexCAN. If a Low-Power request or Freeze mode request is done during a "Permanent Dominant", the corresponding acknowledge can never be asserted.

37.4.10.1 Freeze mode

This mode is requested either by the CPU through the assertion of the HALT bit in the CAN_MCR Register or when the chip is put into Debug mode. In both cases it is also necessary that the FRZ bit is asserted in the CAN_MCR Register and the module is not in a low-power mode.

The acknowledgement is obtained through the assertion by the FlexCAN of FRZ_ACK bit in the same register. The CPU must only consider the FlexCAN in Freeze mode when both request and acknowledgement conditions are satisfied.

When Freeze mode is requested, FlexCAN does the following:

- Waits to be in either Intermission, Passive Error, Bus Off or Idle state
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in does not prevent going to Freeze mode.
- Ignores the Rx input pin and drives the Tx pin as recessive
- Stops the prescaler, thus halting all CAN protocol activities
- Grants write access to the Error Counters Register, which is read-only in other modes
- Sets the NOT_RDY and FRZ_ACK bits in CAN_MCR

After requesting Freeze mode, the user must wait for the FRZ_ACK bit to be asserted in CAN_MCR before executing any other action, otherwise FlexCAN may operate in an unpredictable way. In Freeze mode, all memory mapped registers are accessible, except for CAN_CTRL1[CLKSRC] bit that can be read but cannot be written.

Exiting Freeze mode is done in one of the following ways:

- CPU negates the FRZ bit in the CAN_MCR Register
- The chip is removed from Debug Mode and/or the HALT bit is negated

The FRZ_ACK bit is negated after the protocol engine recognizes the negation of the freeze request. When out of Freeze mode, FlexCAN tries to re-synchronize to the CAN bus by waiting for 11 consecutive recessive bits.

37.4.10.2 Module Disable mode

This low power mode is normally used to temporarily disable a complete FlexCAN block, with no power consumption. It is requested by the CPU through the assertion of the CAN_MCR[MDIS] bit, and the acknowledgement is obtained through the assertion by the FlexCAN of the CAN_MCR[LPMACK] bit. The CPU must only consider the FlexCAN in Disable mode when both request and acknowledgement conditions are satisfied.

If the module is disabled during Freeze mode, it requests to disable the clocks to the PE and CHI sub-modules, sets the LPMACK bit and negates the FRZACK bit.

If the module is disabled during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of --
  >Intermission and then checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive
- Shuts down the clocks to the PE and CHI sub-modules
- Sets the NOTRDY and LPMACK bits in CAN_MCR

The Bus Interface Unit continues to operate, enabling the CPU to access memory mapped registers, except the Rx Mailboxes Global Mask Registers, the Rx Buffer 14 Mask Register, the Rx Buffer 15 Mask Register,. The Message Buffers, the Rx Individual Mask Registers, and the reserved words within RAM may not be accessed when the module is in Disable Mode. Exiting from this mode is done by negating the MDIS bit by the CPU, which causes the FlexCAN to request to resume the clocks and negate the LPMACK bit after the CAN protocol engine recognizes the negation of disable mode requested by the CPU.

37.4.10.3 Doze mode

This is a system low power mode in which the CPU bus is kept alive and a global Doze mode request is sent to all peripherals asking them to enter low-power mode. When Doze mode is globally requested, the DOZE bit in CAN_MCR Register needs to have been asserted previously for Doze mode to be triggered. The acknowledgement is obtained through the assertion by the FlexCAN of the LPMACK bit in the same register. The CPU must only consider the FlexCAN in Doze mode when both request and acknowledgement conditions are satisfied.

If Doze mode is triggered during Freeze mode, FlexCAN requests to shut down the clocks to the PE and CHI sub-modules, sets the LPMACK bit and negates the FRZACK bit.If Doze Mode is triggered during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive
- Shuts down the clocks to the PE and CHI sub-modules
- Sets the NOTRDY and LPMACK bits in CAN_MCR

The Bus Interface Unit continues to operate, enabling the CPU to access memory mapped registers, except the Rx Mailboxes Global Mask Registers, the Rx Buffer 14 Mask Register, the Rx Buffer 15 Mask Register. The the Message Buffers, the Rx Individual Mask Registers, and the reserved words within RAM may not be accessed when the module is in Doze Mode.

Exiting Doze mode is done in one of the following ways:

- CPU removing the Doze mode request
- CPU negating the DOZE bit of the CAN_MCR Register
- Self Wake mechanism

In the Self Wake mechanism, if the SLFWAK bit in CAN_MCR Register was set at the time FlexCAN entered Doze mode, then upon detection of a recessive to dominant transition on the CAN bus, FlexCAN negates the DOZE bit, requests to resume its clocks and negates the LPMACK after the CAN protocol engine recognizes the negation of the Doze mode request. It also sets the WAKINT bit in the ESR Register and, if enabled by

the WAKMSK bit in CAN_MCR, generates a Wake Up interrupt to the CPU. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up. The following table details the effect of SLFWAK and WAKMSK upon wake-up from Doze mode.

Table 37-29. Wake-up from Doze mode

SLFWAK	WAKINT	WAKMSK	FlexCAN clocks enabled	Wake-up interrupt generated
0	-	-	No	No
0	-	-	No	No
1	0	0	No	No
1	0	1	No	No
1	1	0	Yes	No
1	1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the Rx CAN input line while in Doze Mode. See the WAKSRC bit in the description of the Module Configuration Register (CAN_MCR). This feature can be used to protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments.

37.4.10.4 Stop mode

This is a system low-power mode in which all chip clocks can be stopped for maximum power savings. The Stop mode is globally requested by the CPU and the acknowledgement is obtained through the assertion by the FlexCAN of a Stop Acknowledgement signal. The CPU must only consider the FlexCAN in Stop mode when both request and acknowledgement conditions are satisfied.

If FlexCAN receives the global Stop mode request during Freeze mode, it sets the LPMACK bit, negates the FRZACK bit and then sends the Stop Acknowledge signal to the CPU, in order to shut down the clocks globally.

If Stop mode is requested during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive

- Sets the NOTRDY and LPMACK bits in CAN_MCR
- Sends a Stop Acknowledge signal to the CPU, so that it can shut down the clocks globally

Stop mode is exited when the CPU resumes the clocks and removes the Stop Mode request. This can be as a result of the Self Wake mechanism.

In the Self Wake mechanism, if the SLFWAK bit in CAN_MCR Register was set at the time FlexCAN entered Stop mode, then upon detection of a recessive to dominant transition on the CAN bus, FlexCAN sets the WAKINT bit in the CAN_ESR Register and, if enabled by the WAKMSK bit in CAN_MCR, generates a Wake Up interrupt to the CPU. Upon receiving the interrupt, the CPU should resume the clocks and remove the Stop mode request. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up. The following table details the effect of SLFWAK and WAKMSK upon wake-up from Stop mode. Note that wake-up from Stop mode only works when both bits are asserted.

After the CAN protocol engine recognizes the negation of the Stop mode request, the FlexCAN negates the LPMACK bit. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up.

Table 37-30. Wake-up from Stop Mode

SLFWAK	WAKINT	WAKMSK	Chip clocks enabled	Wake-up interrupt generated
0	-	-	No	No
0	-	-	No	No
1	0	0	No	No
1	0	1	No	No
1	1	0	No	No
1	1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the Rx CAN input line while in Stop mode. See the WAKSRC bit in in the description of the Module Configuration Register (CAN_MCR). This feature can be used to protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments.

37.4.11 Interrupts

The module has many interrupt sources: interrupts due to message buffers and interrupts due to the ORed interrupts from MBs, Bus Off, Bus Off Done, Error, Error Fast (errors detected in the data phase of CAN FD format messages with the BRS bit set), Wake Up, Tx Warning, and Rx Warning.

Each one of the message buffers can be an interrupt source, if its corresponding IMASK bit is set. There is no distinction between Tx and Rx interrupts for a particular buffer, under the assumption that the buffer is initialized for either transmission or reception. Each of the buffers has an assigned flag bit in the CAN_IFLAG registers. The bit is set when the corresponding buffer completes a successful transfer and is cleared when the CPU writes it to 1 (unless another interrupt is generated at the same time).

Note

It must be guaranteed that the CPU clears only the bit causing the current interrupt. For this reason, bit manipulation instructions (BSET) must not be used to clear interrupt flags. These instructions may cause accidental clearing of interrupt flags which are set after entering the current interrupt service routine.

If the Rx FIFO is enabled (CAN_MCR[RFEN] = 1) and DMA is disabled (CAN_MCR[DMA] = 0), the interrupts corresponding to MBs 0 to 7 have different meanings. Bit 7 of the CAN_IFLAG1 register becomes the "FIFO Overflow" flag; bit 6 becomes the "FIFO Warning" flag, bit 5 becomes the "Frames Available in FIFO" flag and bits 4-0 are unused. See the description of the Interrupt Flags 1 Register (CAN_IFLAG1) for more information.

If both Rx FIFO and DMA are enabled (CAN_MCR[RFEN] and CAN_MCR[DMA] = 1) the FlexCAN does not generate any FIFO interrupt. Bit 5 of the CAN_IFLAG1 register still indicates "Frames Available in FIFO" and generates a DMA request. Bits 7, 6, 4-0 are unused.

CAUTION

FIFO cannot be enabled when CAN FD feature is enabled.

For a combined interrupt where multiple MB interrupt sources are OR'd together, the interrupt is generated when any of the associated MBs (or FIFO, if applicable) generates an interrupt. In this case, the CPU must read the CAN_IFLAG registers to determine which MB or FIFO source caused the interrupt.

The interrupt sources for Bus Off, Bus Off Done, Error, Error Fast, Wake Up, Tx Warning and Rx Warning generate interrupts like the MB interrupt sources, and can be read from CAN_ESR1 register. The Bus Off, Error, Tx Warning, and Rx Warning interrupt mask bits are located in the CAN_CTRL1 Register; the Wake-Up interrupt mask bit is located in the CAN_MCR.

37.4.12 Bus interface

The CPU access to FlexCAN registers are subject to the following rules:

- Unrestricted read and write access to supervisor registers (registers identified with S/U in Table "Module Memory Map" in Supervisor Mode or with S only) results in access error.
- Read and write access to implemented reserved address space results in access error.
- Write access to positions whose bits are all currently read-only results in access error. If at least one of the bits is not read-only then no access error is issued. Write permission to positions or some of their bits can change depending on the mode of operation or transitory state. Refer to register and bit descriptions for details.
- Read and write access to unimplemented address space results in access error.
- Read and write access to RAM located positions during Low Power Mode results in access error.
- It is possible for the RXIMR memory region to be considered as general purpose memory and available for access. There are two ways of doing this:
  a. If CAN_MCR[IRMQ] is cleared, the individual masks (RXIMR) are disabled. In this case the RXIMR memory region is considered as general purpose memory.
  b. If CAN_MCR[MAXMB] is programmed with a value smaller than the available number of MBs, then the unused memory space can be used as general purpose RAM space. Note that reserved words within RAM cannot be used. As an example, suppose FlexCAN's RAM can support up to 16 MBs, CAN_CTRL2[RFFN] is 0x0, and CAN_MCR[MAXMB] is programmed with zero. The maximum number of MBs in this case becomes one. The RAM starts at 0x0080, and the space from 0x0080 to 0x008F is used by the one MB. The memory space from 0x0090 to 0x017F is available. The space between 0x0180 and 0x087F is reserved. The space from 0x0880 to 0x0883 is used by the one Individual Mask and the available memory in the Mask Registers space would be from 0x0884 to 0x08BF. From 0x08C0 through 0x09DF there are reserved words for internal use which cannot be used as general purpose RAM. As a general rule, free memory space for general purpose depends only on MAXMB.

37.5 Initialization/application information

This section provide instructions for initializing the FlexCAN module.

37.5.1 FlexCAN initialization sequence

The FlexCAN module may be reset in three ways:

- Chip level hard reset, which resets all memory mapped registers asynchronously
- SOFTRST bit in MCR, which resets some of the memory mapped registers synchronously. See Table 37-3 to see what registers are affected by soft reset.

Soft reset is synchronous and has to follow an internal request/acknowledge procedure across clock domains. Therefore, it may take some time to fully propagate its effects. The CAN_MCR[SOFTRST] bit remains asserted while soft reset is pending, so software can poll this bit to know when the reset has completed. Also, soft reset can not be applied while clocks are shut down in a low power mode. The low power mode should be exited and the clocks resumed before applying soft reset.

The clock source should be selected while the module is in Disable mode (see CAN_CTRL1[CLKSRC] bit). After the clock source is selected and the module is enabled (CAN_MCR[MDIS] bit negated), FlexCAN automatically goes to Freeze mode. In Freeze mode, FlexCAN is un-synchronized to the CAN bus, the HALT and FRZ bits in CAN_MCR Register are set, the internal state machines are disabled and the FRZACK and NOTRDY bits in the CAN_MCR Register are set. The Tx pin is in recessive state and FlexCAN does not initiate any transmission or reception of CAN frames. Note that the Message Buffers and the Rx Individual Mask Registers are not affected by reset, so they are not automatically initialized.

For any configuration change/initialization it is required that FlexCAN is put into Freeze mode (see Freeze mode). The following is a generic initialization sequence applicable to the FlexCAN module:

- Initialize the Module Configuration Register (CAN_MCR)
  - Enable the individual filtering per MB and reception queue features by setting the IRMQ bit
  - Enable the warning interrupts by setting the WRNEN bit
  - If required, disable frame self reception by setting the SRXDIS bit

  - Enable the Rx FIFO by setting the RFEN bit
  - If Rx FIFO is enabled and DMA is required, set DMA bit
  - Enable the abort mechanism by setting the AEN bit
  - Enable the local priority feature by setting the LPRIOEN bit
- Initialize the Control 1 Register (CAN_CTRL1) and optionally the CAN Bit Timing Register (CAN_CBT). Initialize also the CAN FD CAN Bit Timing Register (CAN_FDCBT).
  - Determine the bit timing parameters: PROPSEG, PSEG1, PSEG2, RJW
  - Optionally determine the bit timing parameters: EPROPSEG, EPSEG1, EPSEG2, ERJW
  - Determine the CAN FD bit timing parameters: FPROPSEG, FPSEG1, FPSEG2, FRJW
  - Determine the bit rate by programming the PRESDIV field and optionally the EPRESDIV field
  - Determine the CAN FD bit rate by programming the FPRESDIV field
  - Determine the internal arbitration mode (LBUF bit)
- Initialize the Message Buffers
  - The Control and Status word of all Message Buffers must be initialized
  - If Rx FIFO was enabled, the ID filter table must be initialized
  - Other entries in each Message Buffer should be initialized as required
- Initialize the Rx Individual Mask Registers (CAN_RXIMRn)
- Set required interrupt mask bits in the CAN_IMASK Registers (for all MB interrupts), in CAN_MCR Register for Wake-Up interrupt and in CAN_CTRL1 / CAN_CTRL2 Registers (for Bus Off and Error interrupts)
- Negate the HALT bit in CAN_MCR

After the last step listed above, FlexCAN attempts to synchronize to the CAN bus.

Chapter 38
Serial Peripheral Interface (SPI)

38.1 Introduction

The serial peripheral interface (SPI) module provides a synchronous serial bus for communication between a chip and an external peripheral device.

38.1.1 Block Diagram

The block diagram of this module is as follows:

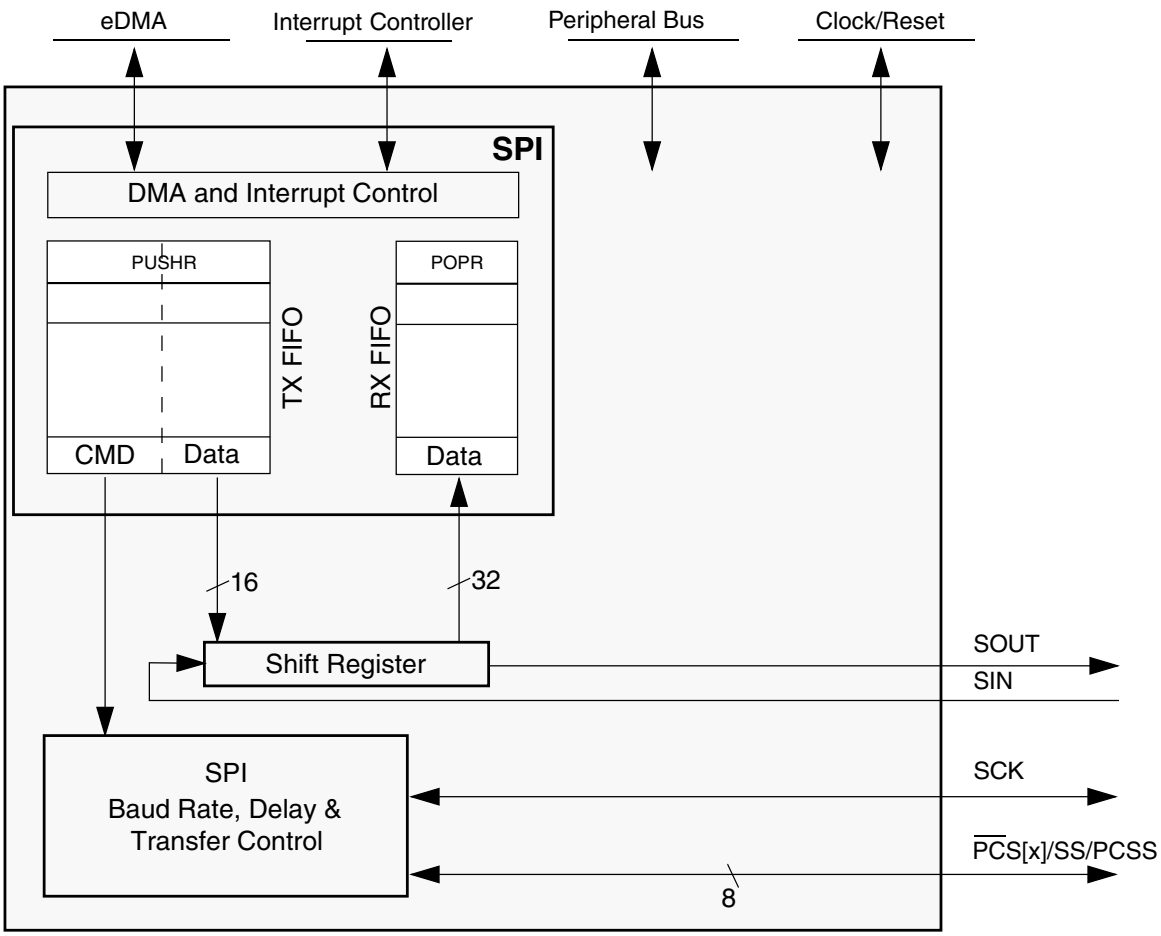


Figure 38-1. SPI Block Diagram

38.1.2 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 324 to 16
    - 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

38.1.3 Interface configurations

38.1.3.1 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.

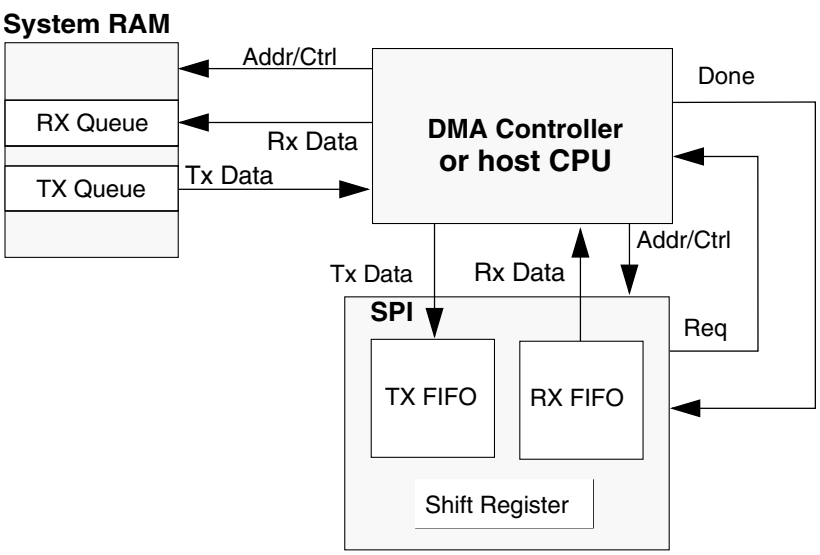


Figure 38-2. SPI with queues and DMA

38.1.4 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.

38.1.4.1 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*]

38.1.4.2 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.

38.1.4.3 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.

38.1.4.4 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.

NOTE

In master mode, 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. In slave mode, this module waits until its chip select input signal goes high before it is ready to have its clocks shut off. It should be noted that the CS0 pad must be correctly configured to allow it to return to high when no transfer is occurring. If the pad is tied low then this could prevent the module from being correctly disabled.

38.1.4.5 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.

38.2 Module signal descriptions

This table describes the signals on the boundary of the module that may connect off chip (in alphabetical order).

Table 38-1. Module signal descriptions

Signal	Master mode	Slave mode	I/O
PCS0/$\overline{SS}$	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

38.2.1 PCS0/$\overline{SS}$—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.

NOTE

Do not tie the SPI slave select pin to ground. Otherwise, SPI cannot function properly.

38.2.2 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

38.2.3 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.

38.2.4 SIN—Serial Input

Master mode: Serial Input (I)—Receives serial data.

Slave mode: Serial Input (I)—Receives serial data.

38.2.5 SOUT—Serial Output

Master mode: Serial Output (O)—Transmits serial data.

Slave mode: Serial Output (O)—Transmits serial data.

38.3 Memory Map/Register Definition

Register accesses to memory addresses that are reserved or undefined result in a transfer error. Any 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	38.3.1/998
4002_C008	Transfer Count Register (SPI0_TCR)	32	R/W	0000_0000h	38.3.2/1001
4002_C00C	Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR0)	32	R/W	7800_0000h	38.3.3/1002
4002_C00C	Clock and Transfer Attributes Register (In Slave Mode) (SPI0_CTAR0_SLAVE)	32	R/W	7800_0000h	38.3.4/1007
4002_C010	Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR1)	32	R/W	7800_0000h	38.3.3/1002
4002_C02C	Status Register (SPI0_SR)	32	R/W	0200_0000h	38.3.5/1008
4002_C030	DMA/Interrupt Request Select and Enable Register (SPI0_RSER)	32	R/W	0000_0000h	38.3.6/1011
4002_C034	PUSH TX FIFO Register In Master Mode (SPI0_PUSHR)	32	R/W	0000_0000h	38.3.7/1013
4002_C034	PUSH TX FIFO Register In Slave Mode (SPI0_PUSHR_SLAVE)	32	R/W	0000_0000h	38.3.8/1015
4002_C038	POP RX FIFO Register (SPI0_POPR)	32	R	0000_0000h	38.3.9/1015
4002_C03C	Transmit FIFO Registers (SPI0_TXFR0)	32	R	0000_0000h	38.3.10/ 1016
4002_C040	Transmit FIFO Registers (SPI0_TXFR1)	32	R	0000_0000h	38.3.10/ 1016
4002_C044	Transmit FIFO Registers (SPI0_TXFR2)	32	R	0000_0000h	38.3.10/ 1016
4002_C048	Transmit FIFO Registers (SPI0_TXFR3)	32	R	0000_0000h	38.3.10/ 1016
4002_C07C	Receive FIFO Registers (SPI0_RXFR0)	32	R	0000_0000h	38.3.11/ 1016
4002_C080	Receive FIFO Registers (SPI0_RXFR1)	32	R	0000_0000h	38.3.11/ 1016
4002_C084	Receive FIFO Registers (SPI0_RXFR2)	32	R	0000_0000h	38.3.11/ 1016
4002_C088	Receive FIFO Registers (SPI0_RXFR3)	32	R	0000_0000h	38.3.11/ 1016
4002_D000	Module Configuration Register (SPI1_MCR)	32	R/W	0000_4001h	38.3.1/998
4002_D008	Transfer Count Register (SPI1_TCR)	32	R/W	0000_0000h	38.3.2/1001
4002_D00C	Clock and Transfer Attributes Register (In Master Mode) (SPI1_CTAR0)	32	R/W	7800_0000h	38.3.3/1002

Table continues on the next page...

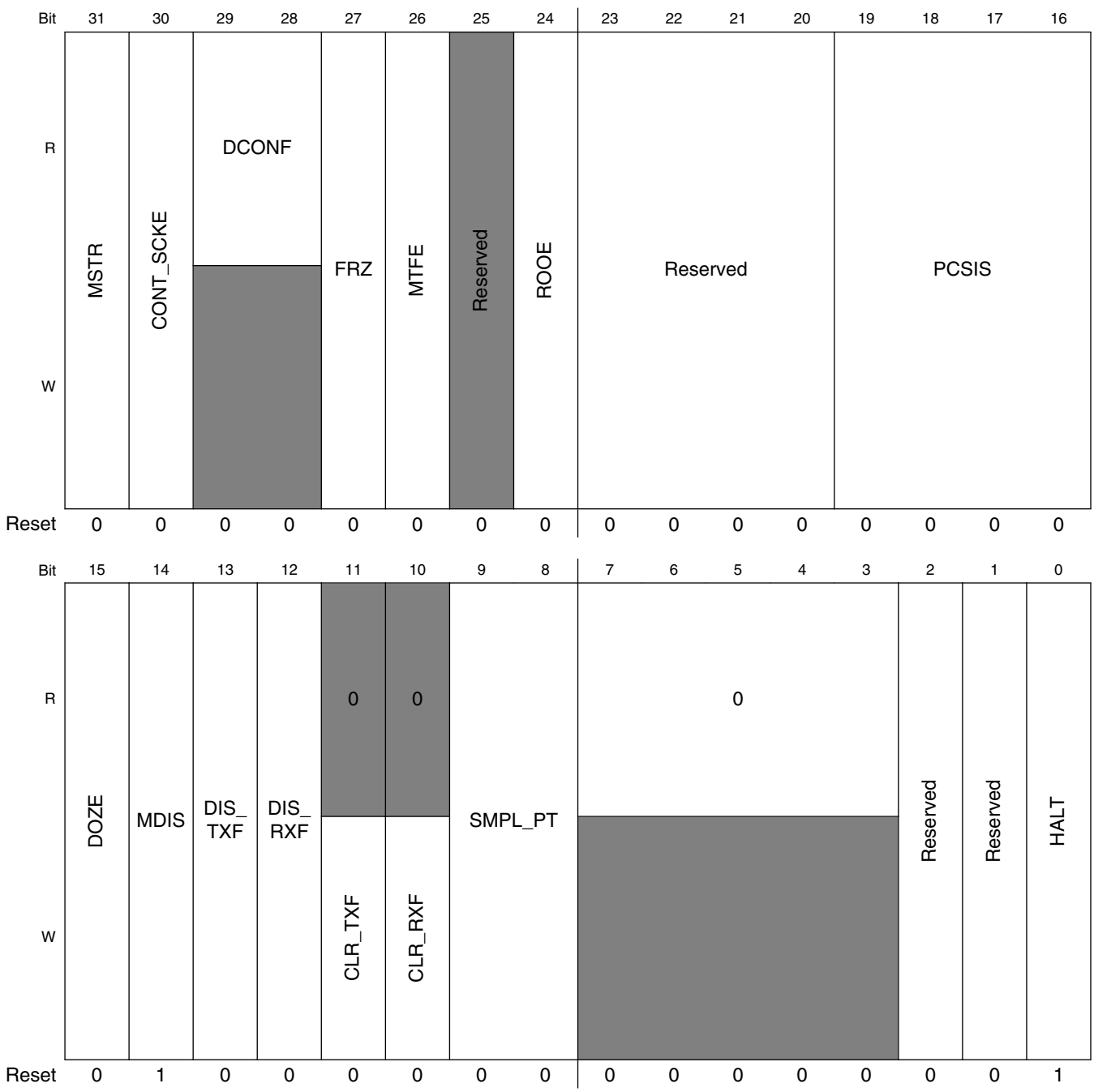
SPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_D00C	Clock and Transfer Attributes Register (In Slave Mode) (SPI1_CTAR0_SLAVE)	32	R/W	7800_0000h	38.3.4/1007
4002_D010	Clock and Transfer Attributes Register (In Master Mode) (SPI1_CTAR1)	32	R/W	7800_0000h	38.3.3/1002
4002_D02C	Status Register (SPI1_SR)	32	R/W	0200_0000h	38.3.5/1008
4002_D030	DMA/Interrupt Request Select and Enable Register (SPI1_RSER)	32	R/W	0000_0000h	38.3.6/1011
4002_D034	PUSH TX FIFO Register In Master Mode (SPI1_PUSHR)	32	R/W	0000_0000h	38.3.7/1013
4002_D034	PUSH TX FIFO Register In Slave Mode (SPI1_PUSHR_SLAVE)	32	R/W	0000_0000h	38.3.8/1015
4002_D038	POP RX FIFO Register (SPI1_POPR)	32	R	0000_0000h	38.3.9/1015
4002_D03C	Transmit FIFO Registers (SPI1_TXFR0)	32	R	0000_0000h	38.3.10/ 1016
4002_D040	Transmit FIFO Registers (SPI1_TXFR1)	32	R	0000_0000h	38.3.10/ 1016
4002_D044	Transmit FIFO Registers (SPI1_TXFR2)	32	R	0000_0000h	38.3.10/ 1016
4002_D048	Transmit FIFO Registers (SPI1_TXFR3)	32	R	0000_0000h	38.3.10/ 1016
4002_D07C	Receive FIFO Registers (SPI1_RXFR0)	32	R	0000_0000h	38.3.11/ 1016
4002_D080	Receive FIFO Registers (SPI1_RXFR1)	32	R	0000_0000h	38.3.11/ 1016
4002_D084	Receive FIFO Registers (SPI1_RXFR2)	32	R	0000_0000h	38.3.11/ 1016
4002_D088	Receive FIFO Registers (SPI1_RXFR3)	32	R	0000_0000h	38.3.11/ 1016

38.3.1 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



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. **NOTE:** When MTFE=1 with continuous SCK enabled (MCR [CONT_SCKE] =1) in master mode, configure CTAR[LSBFE]=0 for correct operations while receiving unequal length frames. If PUSHR[CONT] is also set for back to back frame transfer, also configure the frame size of the first frame as less than or equal to the frame size of the next frame. In this scenario, make sure that for all received frames, the bits are read equal to their respective frame sizes and any extra bits during POP operation are masked. 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.

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
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 SPI interface. 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	Clear RX FIFO Flushes the RX FIFO. Writing a 1 to CLR_RXF clears the RX Counter. The CLR_RXF bit is always read as zero.

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
	NOTE: After every RX FIFO clear operation (MCR [CLR_RXF] = 0b1) following a RX FIFO overflow (SR [RFOF] = 0b1) scenario, perform a single POP from the RX FIFO and discard the read data. The POP and discard operation must be completed before receiving a new incoming frame. 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 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.

38.3.2 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

Table continues on the next page...

SPIx_TCR field descriptions (continued)

Field	Description
	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.

38.3.3 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 a 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

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>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 38-2. SPI SCK Duty Cycle
<table><tr><th>DBR</th><th>CPHA</th><th>PBR</th><th>SCK Duty Cycle</th></tr><tr><td>0</td><td>any</td><td>any</td><td>50/50</td></tr><tr><td>1</td><td>0</td><td>00</td><td>50/50</td></tr><tr><td>1</td><td>0</td><td>01</td><td>33/66</td></tr><tr><td>1</td><td>0</td><td>10</td><td>40/60</td></tr><tr><td>1</td><td>0</td><td>11</td><td>43/57</td></tr><tr><td>1</td><td>1</td><td>00</td><td>50/50</td></tr><tr><td>1</td><td>1</td><td>01</td><td>66/33</td></tr><tr><td>1</td><td>1</td><td>10</td><td>60/40</td></tr><tr><td>1</td><td>1</td><td>11</td><td>57/43</td></tr></table>
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.</td></tr>
<tr><td>30–27
FMSZ</td><td>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.</td></tr>
<tr><td>26
CPOL</td><td>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 guaranteed.

0 The inactive state value of SCK is low.
1 The inactive state value of SCK is high.</td></tr>
<tr><td>25
CPHA</td><td>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.</td></tr>
<tr><td>24
LSBFE</td><td>LSB First

Specifies whether the LSB or MSB of the frame is transferred first.</td></tr>
</table>

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

Field	Description
	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. 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 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 38-3. Delay Scaler Encoding
<table>
<tr><th>Field Value</th><th>Delay 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>8</td></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:

$t_{ASC} = (1/f_P) \times PASC \times ASC$

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:

$t_{DT} = (1/f_P) \times PDT \times DT$

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:</td></tr>
</table>

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

Field	Description
	SCK baud rate = (f_P /PBR) x [(1+DBR)/BR] The following table lists the baud rate scaler values.

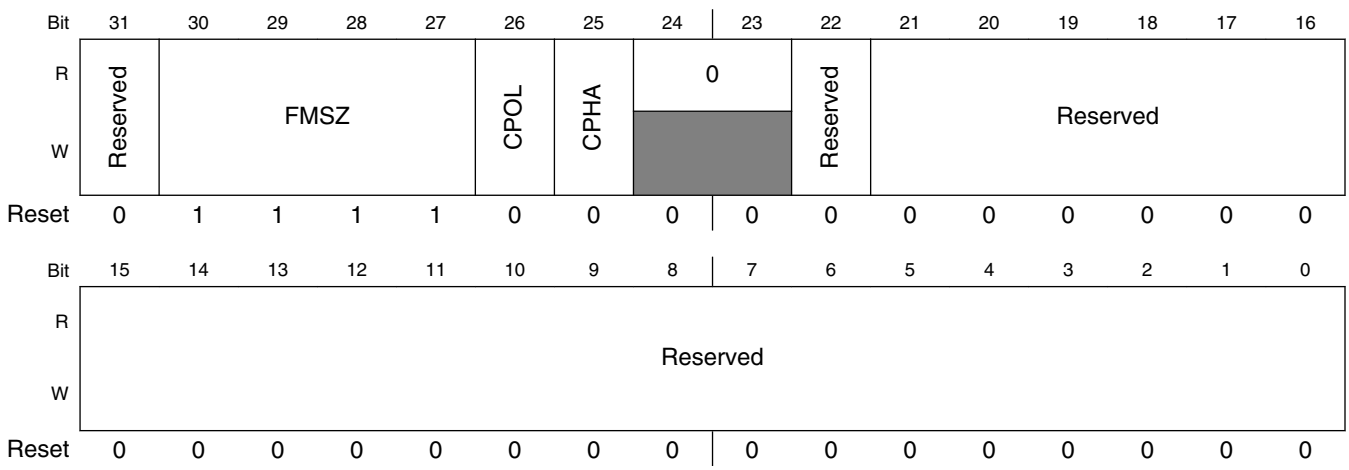
Table 38-4. Baud Rate Scaler

CTARn[BR]	Baud Rate Scaler Value
0000	2
0001	4
0010	6
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
1010	1024
1011	2048
1100	4096
1101	8192
1110	16384
1111	32768

38.3.4 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



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 transferred 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 guaranteed. 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.

Table continues on the next page...

SPIx_CTAR*n*_SLAVE field descriptions (continued)

Field	Description
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.

38.3.5 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	
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.
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 Indicates whether there is an available location to be filled in the FIFO. Either a DMA request or an interrupt indication can be used to add another entry to the FIFO. Note that this bit is set if at least one location is free in the FIFO. The TFFF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller to the TX FIFO full request, when the TX FIFO is full. **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.

Table continues on the next page...

SPIx_SR field descriptions (continued)

Field	Description
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 Indicates whether there is an available location to be drained from the FIFO. Either a DMA request or an interrupt indication can be used to read from the FIFO. Note that this bit is set if at least one location can be read from the FIFO. 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.

38.3.6 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.

38.3.7 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.

38.3.8 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.

38.3.9 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

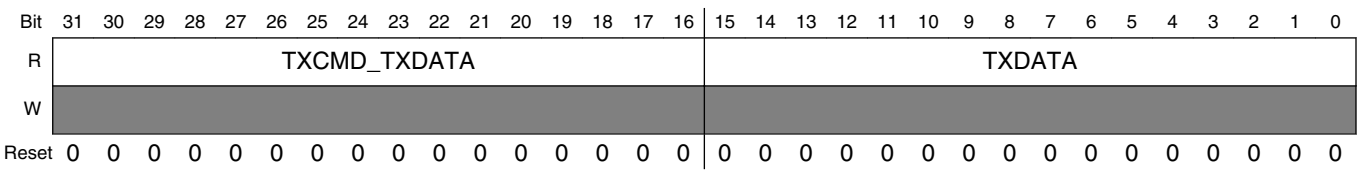
SPIx_POPR field descriptions

Field	Description
RXDATA	Received Data Contains the SPI data from the RX FIFO entry to which the Pop Next Data Pointer points.

38.3.10 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



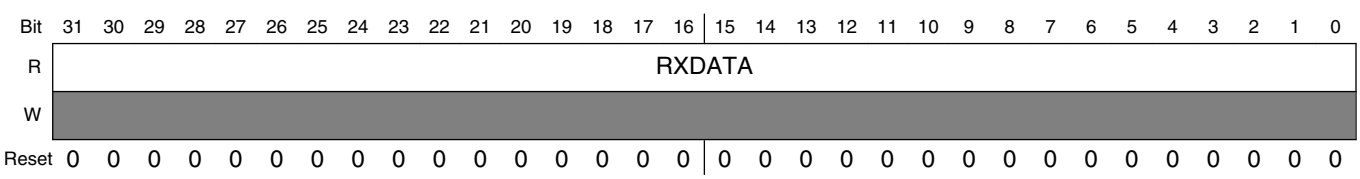
SPIx_TXFR*n* field descriptions

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.

38.3.11 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



SPIx_RXFR*n* field descriptions

Field	Description
RXDATA	Receive Data Contains the received SPI data.

SPIx_RXFR*n* field descriptions (continued)

Field	Description

38.4 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.

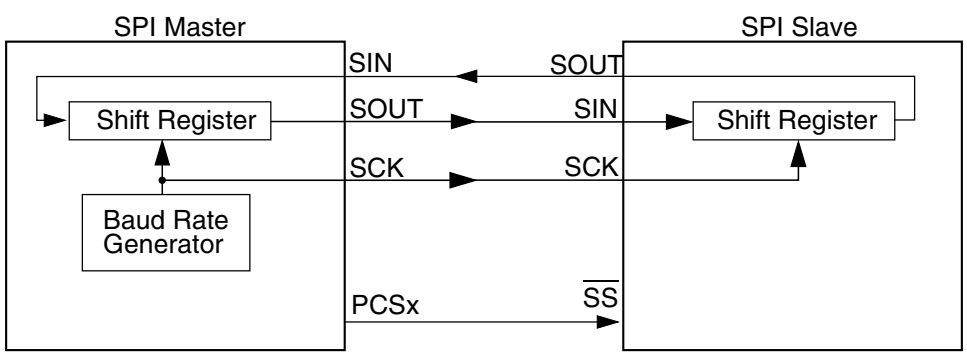


Figure 38-3. Serial protocol overview

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.

38.4.1 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.

38.4.2 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
- 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.

38.4.2.1 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.

38.4.2.2 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.

38.4.2.3 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.

38.4.2.4 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.

38.4.2.4.1 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.

38.4.2.4.2 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.

38.4.2.5 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.

38.4.2.5.1 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.

38.4.2.5.2 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.

38.4.3 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.

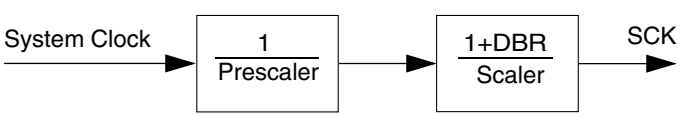


Figure 38-4. Communications clock prescalers and scalers

38.4.3.1 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 38-5. 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.

38.4.3.2 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 38-5 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 38-6. 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.

38.4.3.3 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 38-5 and Figure 38-6 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 38-7. 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.

38.4.3.4 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 38-5 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 38-8. 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.

38.4.4 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.

38.4.4.1 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.

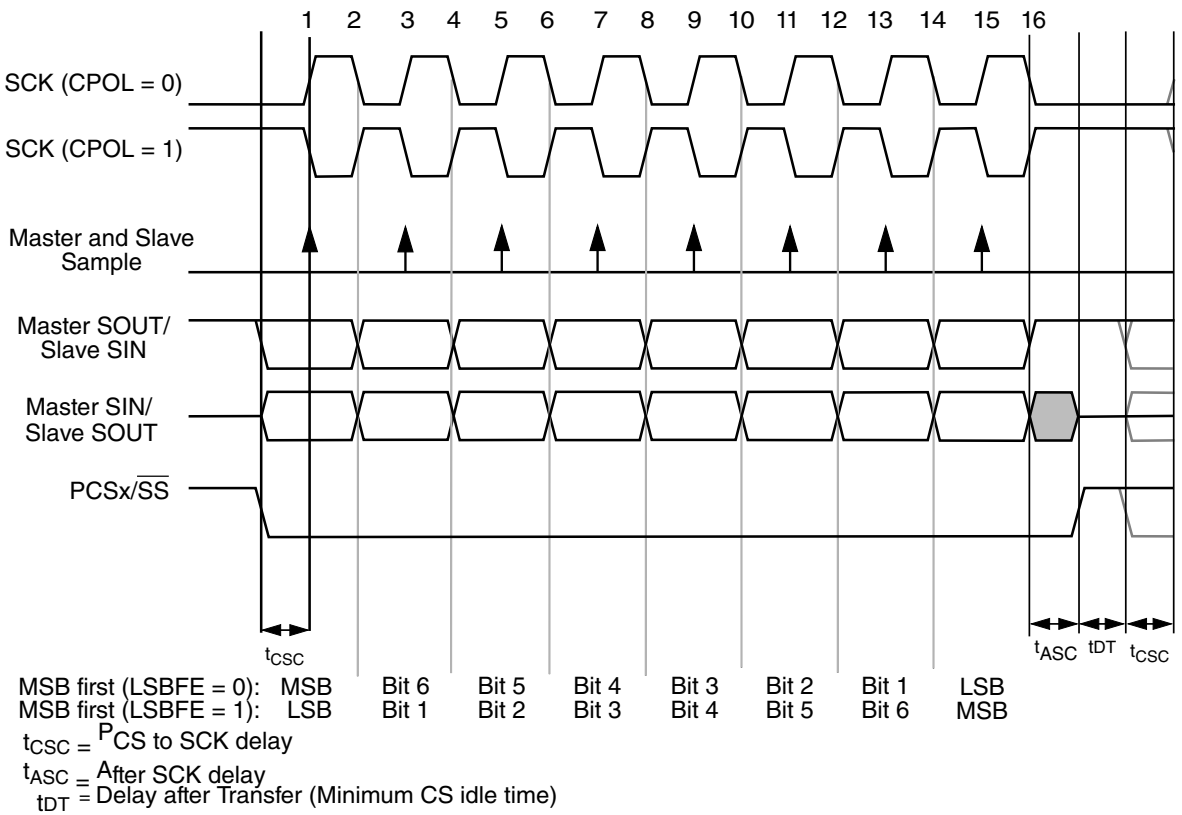


Figure 38-5. Module transfer timing diagram (MTFE=0, CPHA=0, FMSZ=8)

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.

38.4.4.2 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.

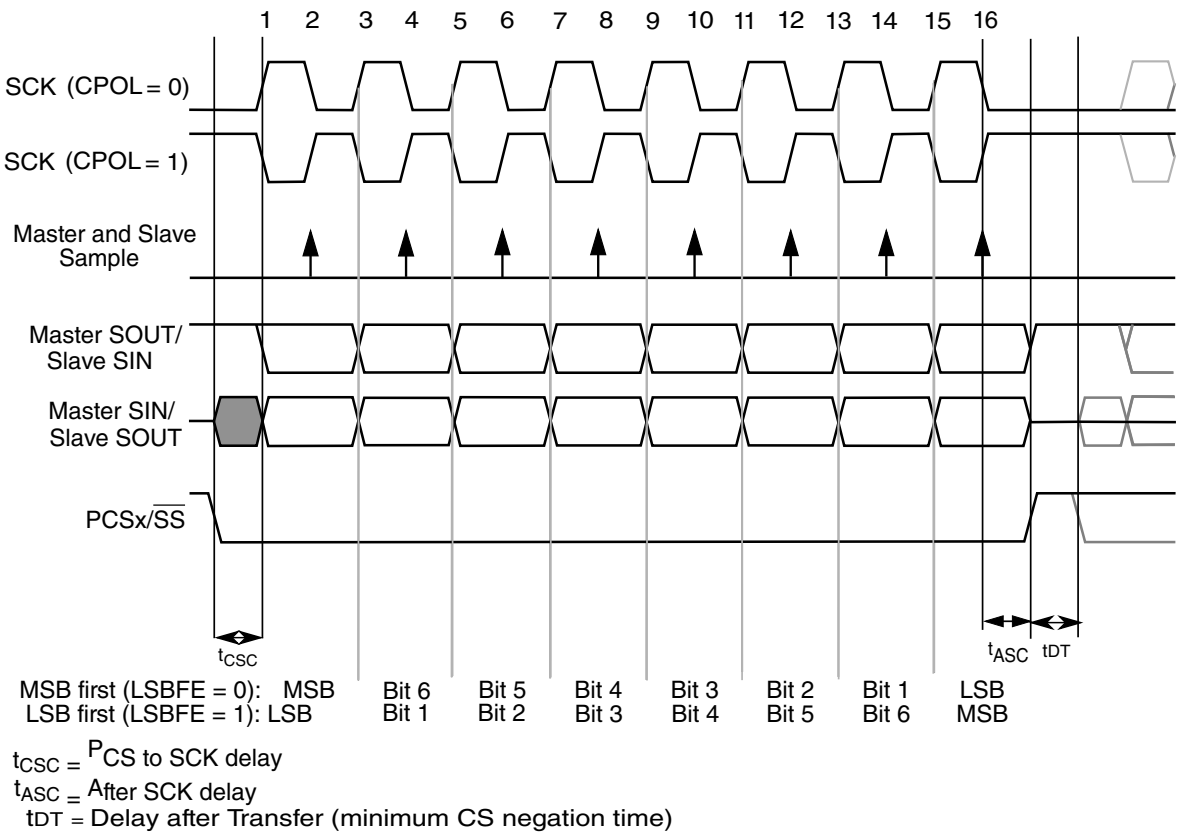


Figure 38-6. Module transfer timing diagram (MTFE=0, CPHA=1, FMSZ=8)

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.

38.4.4.3 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 SPI 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 SPI 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 SPI_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 SPI 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 SPI Slave", presents SPI in the slave mode with MTFE bit set.

Other MTFE = 1 diagrams show SPI SIN input as being driven by a regular external SPI slave, configured according SPI master CPHA programming.

Note

In the following diagrams, f_{sys} represents the protocol clock frequency from which the Baud frequency f_{sck} is derived.

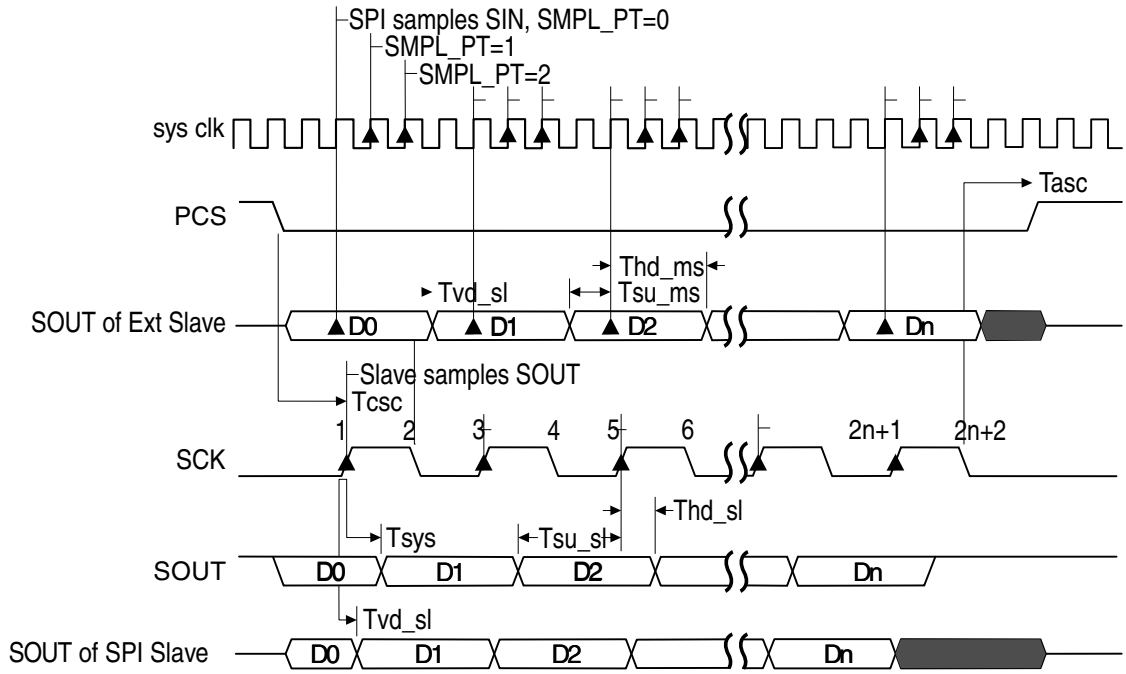


Figure 38-7. SPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/4$)

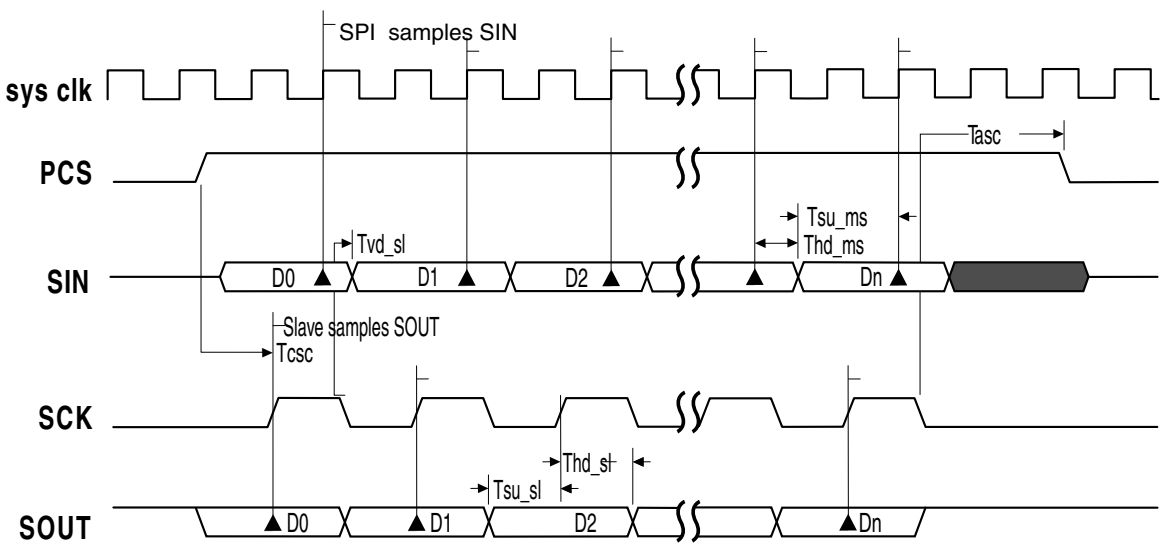


Figure 38-8. SPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/2$)

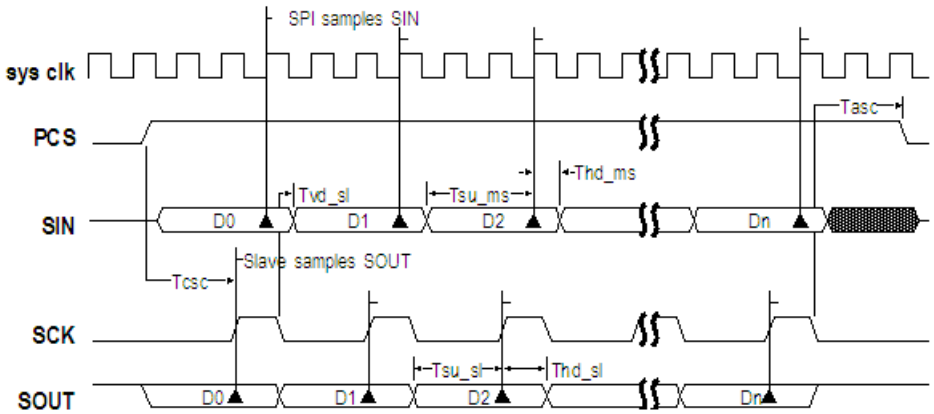


Figure 38-9. SPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/3$)

38.4.4.4 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 SPI 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.**

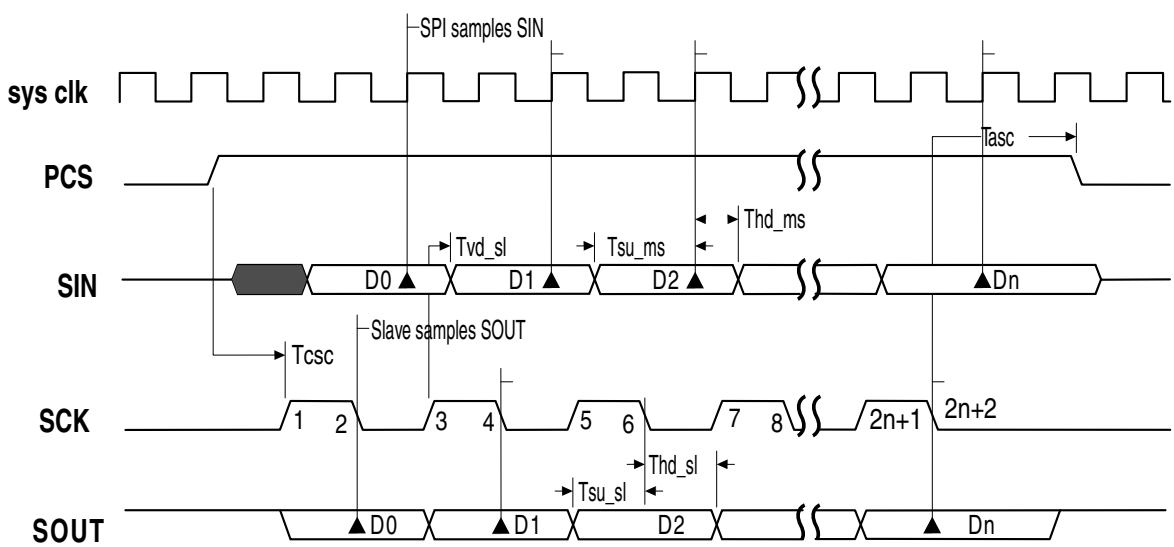


Figure 38-10. SPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/2$)

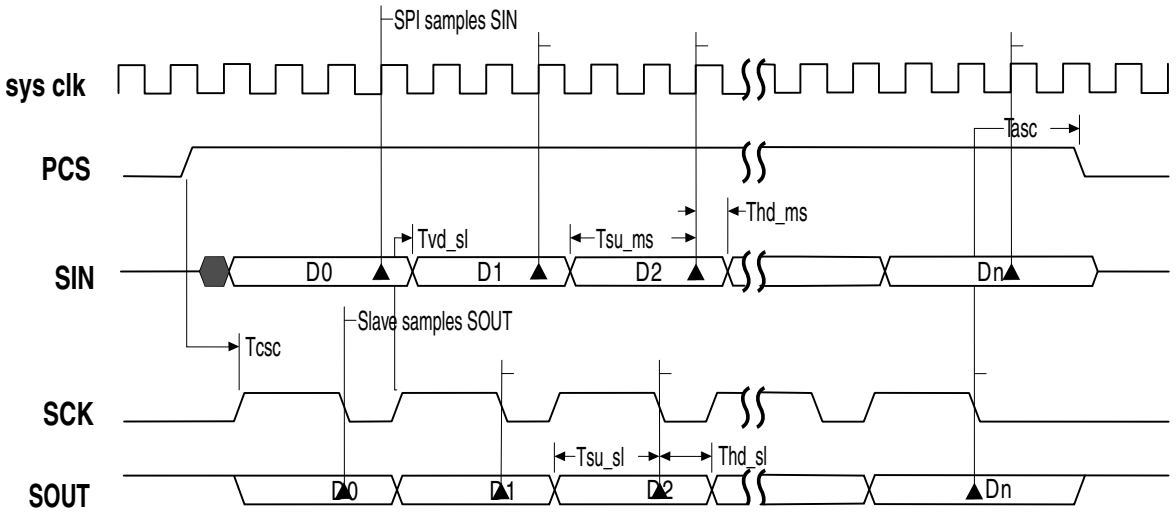


Figure 38-11. SPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/3$)

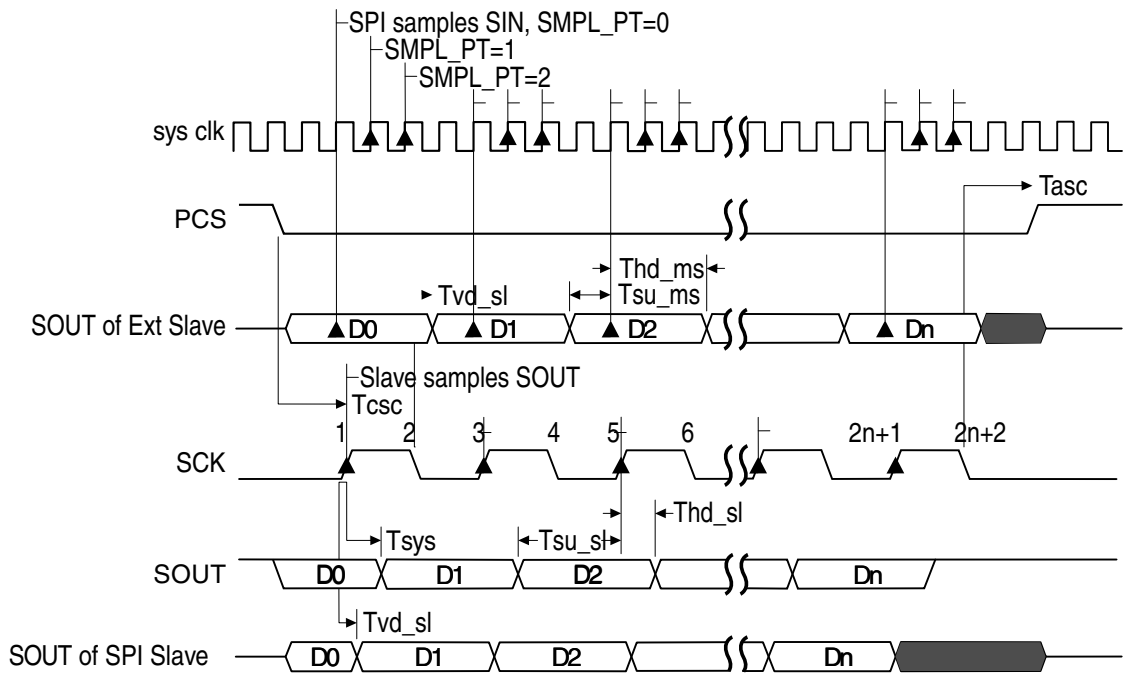


Figure 38-12. SPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/4$)

38.4.4.5 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.

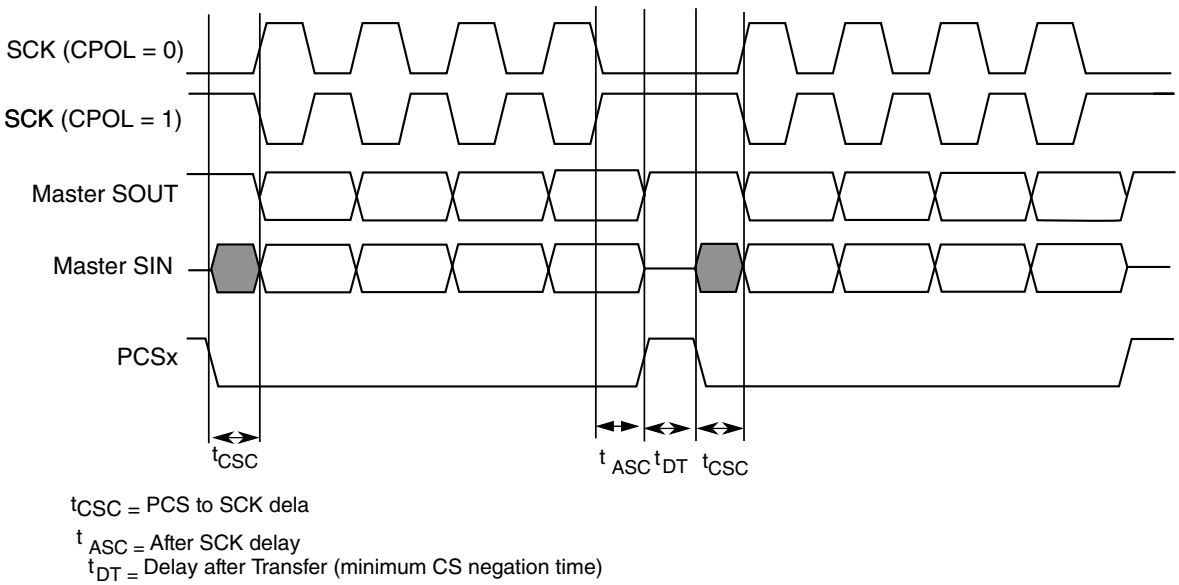


Figure 38-13. Example of non-continuous format (CPHA=1, CONT=0)

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.

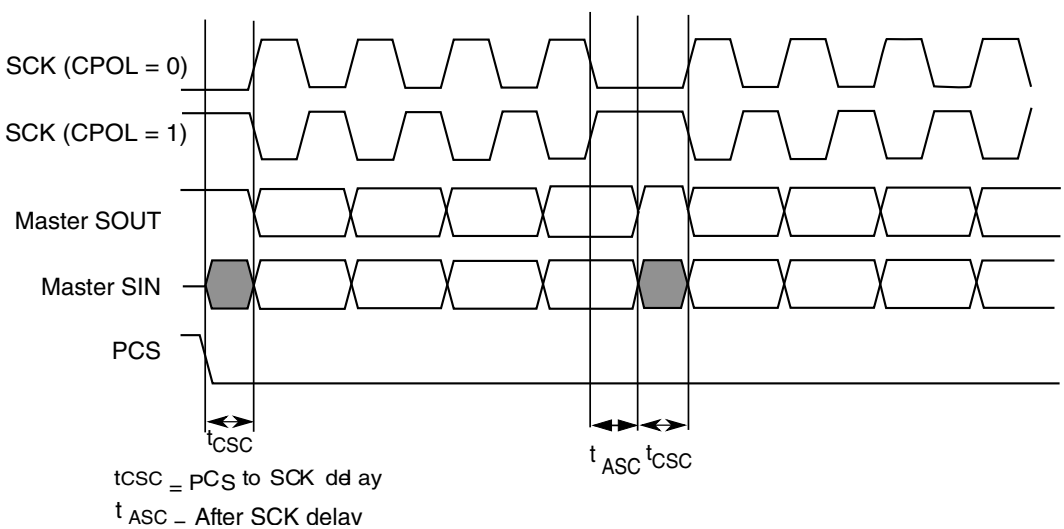


Figure 38-14. Example of continuous transfer (CPHA=1, CONT=1)

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.

38.4.5 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.

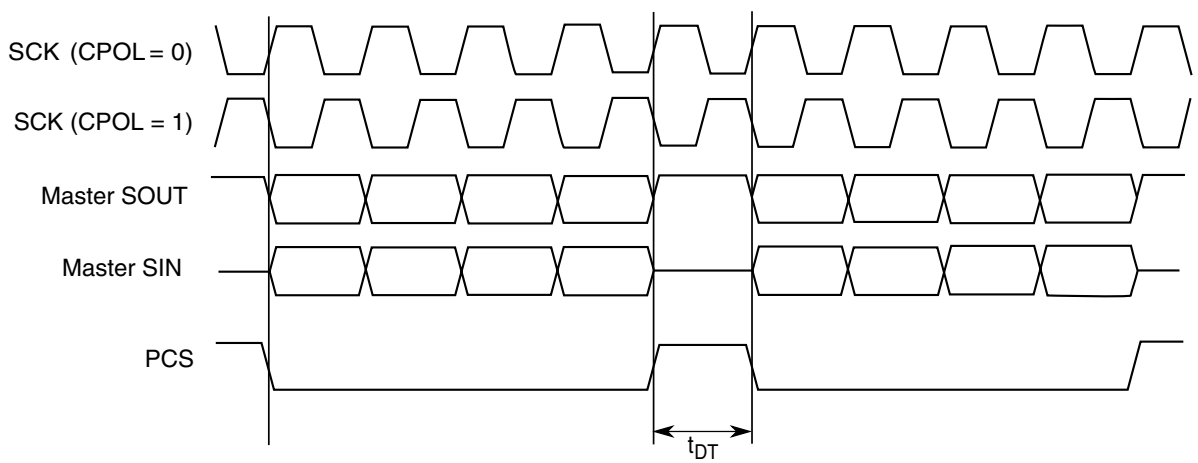


Figure 38-15. Continuous SCK Timing Diagram (CONT=0)

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.

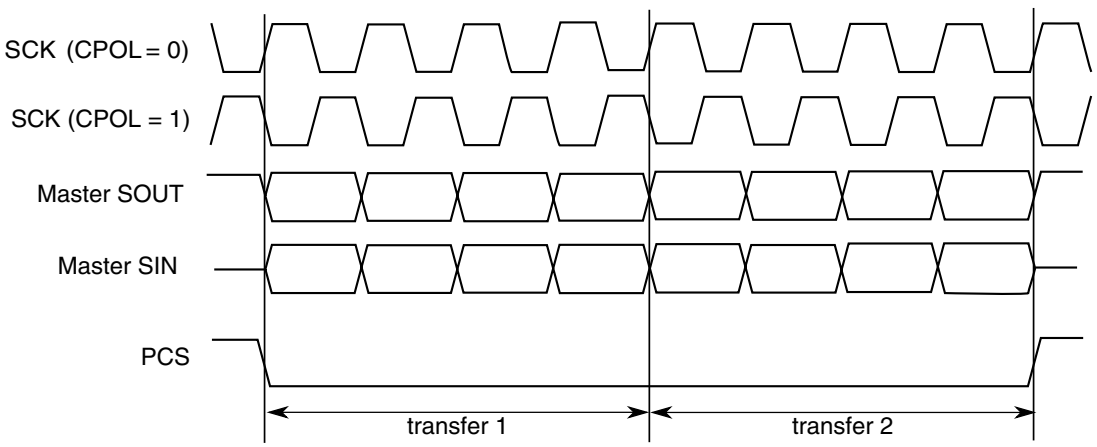


Figure 38-16. Continuous SCK timing diagram (CONT=1)

38.4.6 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 $\overline{SS}$ signal is asserted and any time when transmit data is ready and $\overline{SS}$ 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 $\overline{SS}$ negates before that last SCK edge, the data from shift register is lost.

38.4.7 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 38-9. 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	-
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.

38.4.7.1 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.

38.4.7.2 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. Configure the DMA to fill only one FIFO location per transfer.

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.

38.4.7.3 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.

38.4.7.4 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.

38.4.7.5 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. Configure the DMA to drain only one FIFO location per transfer.

38.4.7.6 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.

38.4.8 Power saving features

The module supports following power-saving strategies:

- External Stop mode
- Module Disable mode – Clock gating of non-memory mapped logic

38.4.8.1 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.

38.4.8.2 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.

38.5 Initialization/application information

This section describes how to initialize the module.

38.5.1 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.

38.5.2 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].

38.5.3 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.

38.5.4 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 38-10. 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

38.5.5 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 38-11. 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

38.5.6 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.

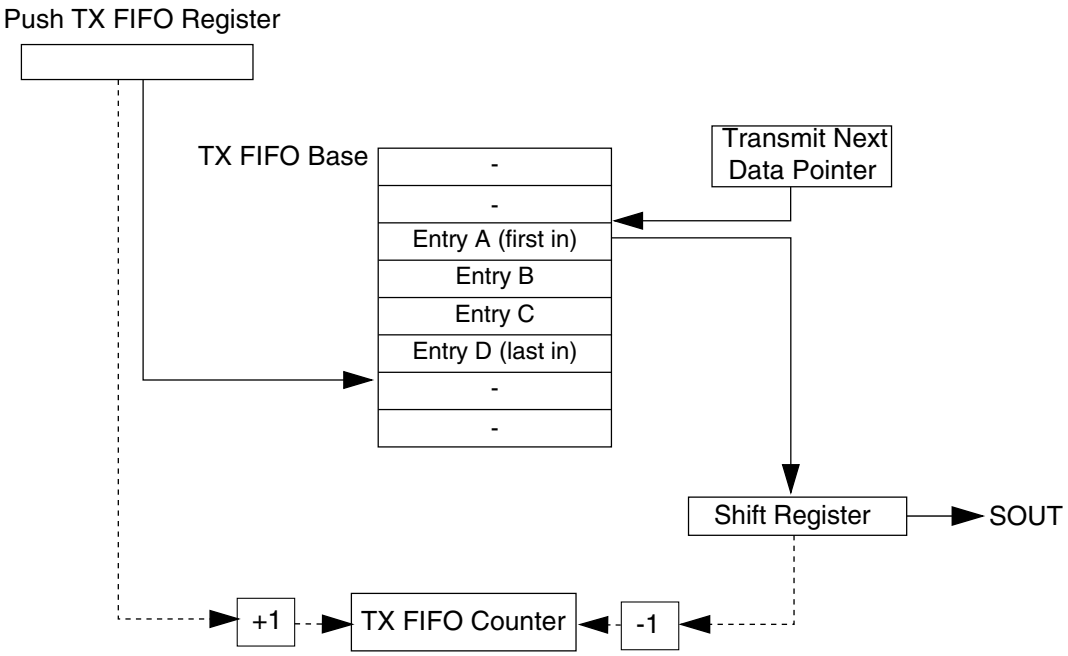


Figure 38-17. TX FIFO pointers and counter

38.5.6.1 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:

$$\text{First-in EntryAddress} = \text{TXFIFOBase} + \left(4 \times \text{TXNXTPTR}\right)$$

The memory address of the last-in entry in the TX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{TXFIFOBase} + 4 \times \left(\text{TXCTR} + \text{TXNXTPTR} - 1\right) \text{mod} \left(\text{TXFIFOdepth}\right)$$

TX FIFO Base - Base address of TX FIFO
TXCTR - TX FIFO Counter
TXNXTPTR - Transmit Next Pointer
TX FIFO Depth - Transmit FIFO depth, implementation specific

38.5.6.2 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 39
Inter-Integrated Circuit (I2C)

The inter-integrated circuit (I^2C, I2C, or IIC) module provides a method of communication between a number of devices.

39.1 Introduction

The inter-integrated circuit (I^2C, 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*.

39.1.1 Features

The I2C module has the following features:

- Compatible with *The I^2C-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

39.1.2 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.

39.1.3 Block diagram

The following figure is a functional block diagram of the I2C module.

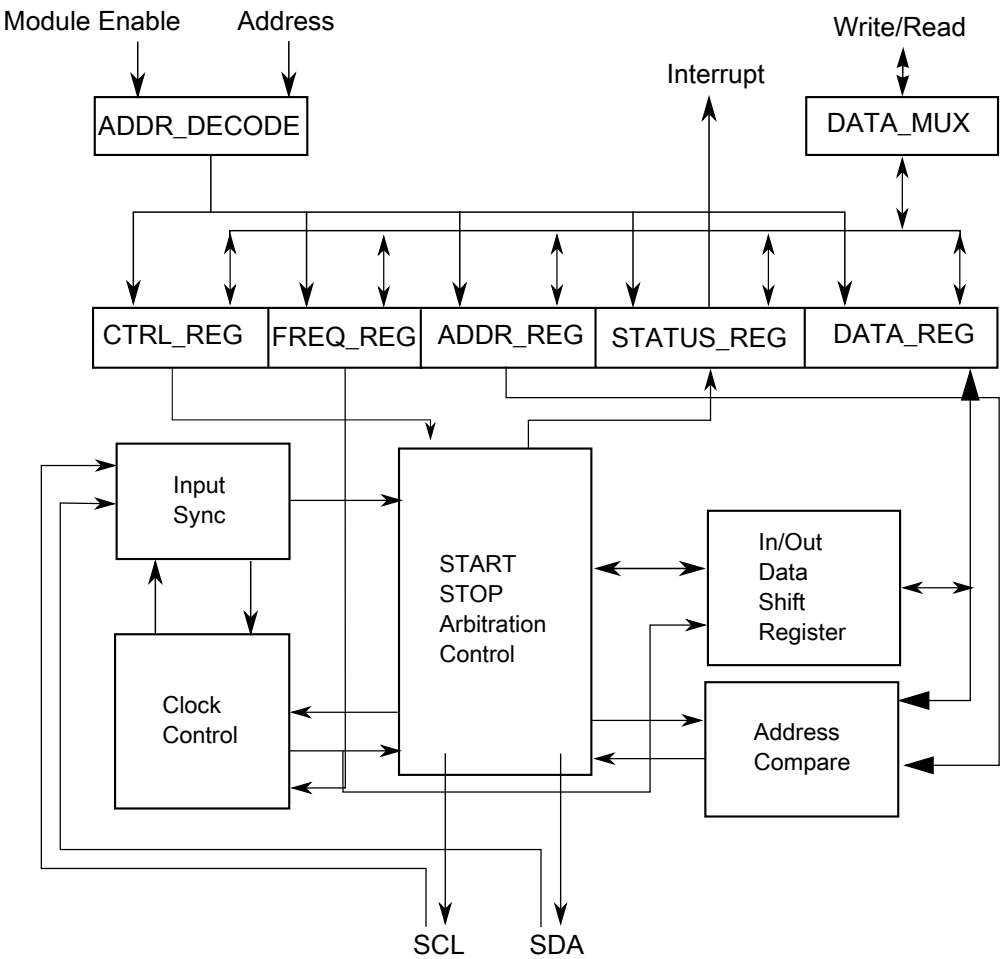


Figure 39-1. I2C Functional block diagram

39.2 I²C signal descriptions

The signal properties of I²C are shown in the table found here.

Table 39-1. I²C signal descriptions

Signal	Description	I/O
SCL	Bidirectional serial clock line of the I²C system.	I/O
SDA	Bidirectional serial data line of the I²C system.	I/O

39.3 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	39.3.1/1049
4006_6001	I2C Frequency Divider register (I2C0_F)	8	R/W	00h	39.3.2/1049
4006_6002	I2C Control Register 1 (I2C0_C1)	8	R/W	00h	39.3.3/1050
4006_6003	I2C Status register (I2C0_S)	8	R/W	80h	39.3.4/1052
4006_6004	I2C Data I/O register (I2C0_D)	8	R/W	00h	39.3.5/1054
4006_6005	I2C Control Register 2 (I2C0_C2)	8	R/W	00h	39.3.6/1054
4006_6006	I2C Programmable Input Glitch Filter Register (I2C0_FLT)	8	R/W	00h	39.3.7/1055
4006_6007	I2C Range Address register (I2C0_RA)	8	R/W	00h	39.3.8/1057
4006_6008	I2C SMBus Control and Status register (I2C0_SMB)	8	R/W	00h	39.3.9/1057
4006_6009	I2C Address Register 2 (I2C0_A2)	8	R/W	C2h	39.3.10/ 1059
4006_600A	I2C SCL Low Timeout Register High (I2C0_SLTH)	8	R/W	00h	39.3.11/ 1059
4006_600B	I2C SCL Low Timeout Register Low (I2C0_SLTL)	8	R/W	00h	39.3.12/ 1060
4006_600C	I2C Status register 2 (I2C0_S2)	8	R/W	01h	39.3.13/ 1060
4006_7000	I2C Address Register 1 (I2C1_A1)	8	R/W	00h	39.3.1/1049
4006_7001	I2C Frequency Divider register (I2C1_F)	8	R/W	00h	39.3.2/1049
4006_7002	I2C Control Register 1 (I2C1_C1)	8	R/W	00h	39.3.3/1050
4006_7003	I2C Status register (I2C1_S)	8	R/W	80h	39.3.4/1052
4006_7004	I2C Data I/O register (I2C1_D)	8	R/W	00h	39.3.5/1054
4006_7005	I2C Control Register 2 (I2C1_C2)	8	R/W	00h	39.3.6/1054
4006_7006	I2C Programmable Input Glitch Filter Register (I2C1_FLT)	8	R/W	00h	39.3.7/1055
4006_7007	I2C Range Address register (I2C1_RA)	8	R/W	00h	39.3.8/1057
4006_7008	I2C SMBus Control and Status register (I2C1_SMB)	8	R/W	00h	39.3.9/1057
4006_7009	I2C Address Register 2 (I2C1_A2)	8	R/W	C2h	39.3.10/ 1059
4006_700A	I2C SCL Low Timeout Register High (I2C1_SLTH)	8	R/W	00h	39.3.11/ 1059
4006_700B	I2C SCL Low Timeout Register Low (I2C1_SLTL)	8	R/W	00h	39.3.12/ 1060
4006_700C	I2C Status register 2 (I2C1_S2)	8	R/W	01h	39.3.13/ 1060

39.3.1 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.

39.3.2 I2C Frequency Divider register (I2Cx_F)

Address: Base address + 1h offset

Bit	7	6	5	4	3	2	1	0
Read	MULT		ICR					
Write								
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) x mul x 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) x mul x 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) x mul x 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^2C 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

39.3.3 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.

39.3.4 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...

I2C*x*_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

39.3.5 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).

39.3.6 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.

39.3.7 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

39.3.8 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.

39.3.9 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

39.3.10 I2C Address Register 2 (I2Cx_A2)

Address: Base address + 9h offset

Bit	7	6	5	4	3	2	1	0
Read Write	SAD							0
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.

39.3.11 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.

39.3.12 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.

39.3.13 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.

39.4 Functional description

This section provides a comprehensive functional description of the I2C module.

39.4.1 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.

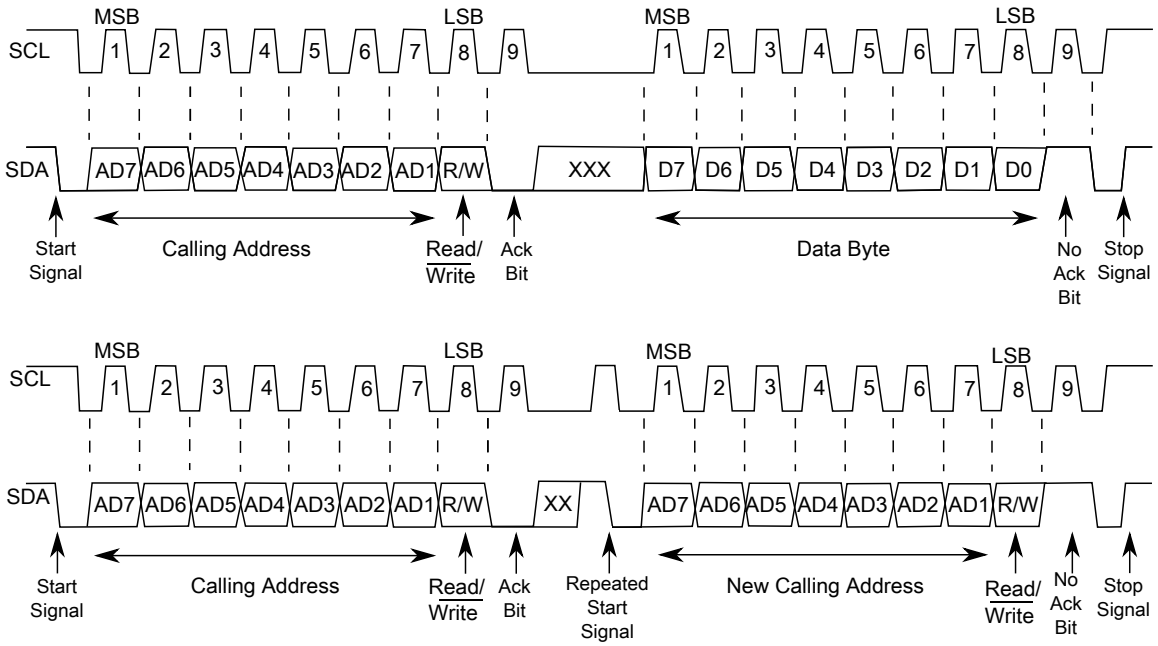


Figure 39-2. I2C bus transmission signals

39.4.1.1 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.

39.4.1.2 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/$\overline{W}$ bit. The R/$\overline{W}$ 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.

39.4.1.3 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.

39.4.1.4 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.

39.4.1.5 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.

39.4.1.6 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.

39.4.1.7 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.

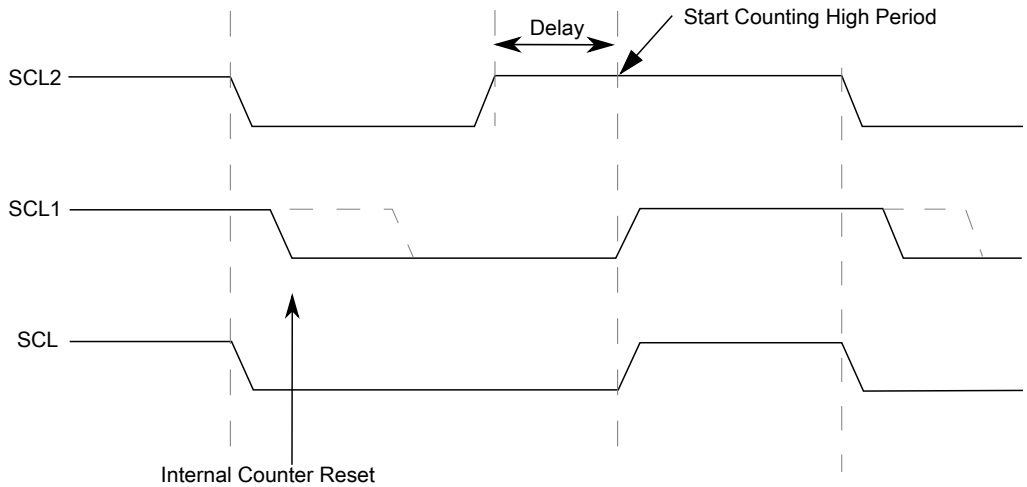


Figure 39-3. I2C clock synchronization

39.4.1.8 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.

39.4.1.9 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.

39.4.1.10 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.

Table 39-2. 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

39.4.2 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.

39.4.2.1 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 39-3. 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.

39.4.2.2 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 39-4. 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.

39.4.3 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.

39.4.4 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.

39.4.4.1 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.

39.4.4.1.1 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.

39.4.4.1.2 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.

39.4.4.1.3 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.

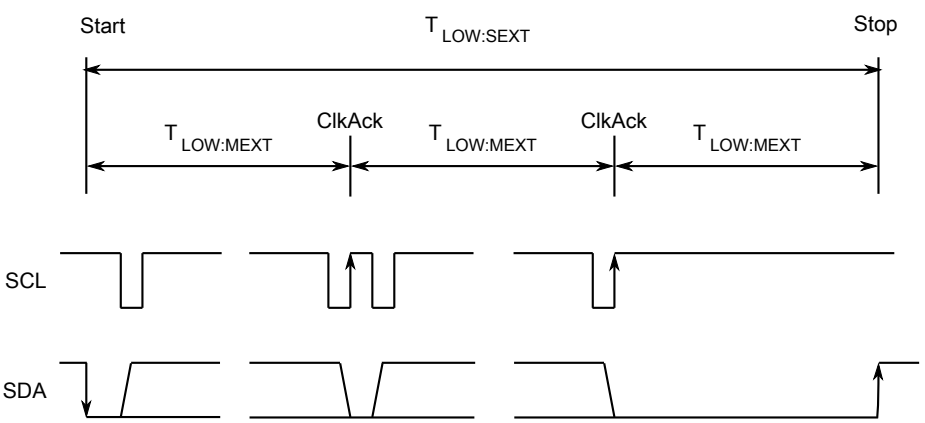


Figure 39-4. Timeout measurement intervals

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.

39.4.4.2 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.

39.4.5 Resets

The I2C module is disabled after a reset. The I2C module cannot cause a core reset.

39.4.6 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 39-5. 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^2C bus stop detection	STOPF	IICIF	IICIE & SSIE
I^2C 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

39.4.6.1 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.

39.4.6.2 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.

39.4.6.3 Stop Detect Interrupt

When the stop status is detected on the I^2C bus, the STOPF bit is set to 1. The CPU is interrupted, provided the IICIE and SSIE bits are both set to 1.

39.4.6.4 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.

39.4.6.5 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.

39.4.6.6 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.

39.4.7 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.

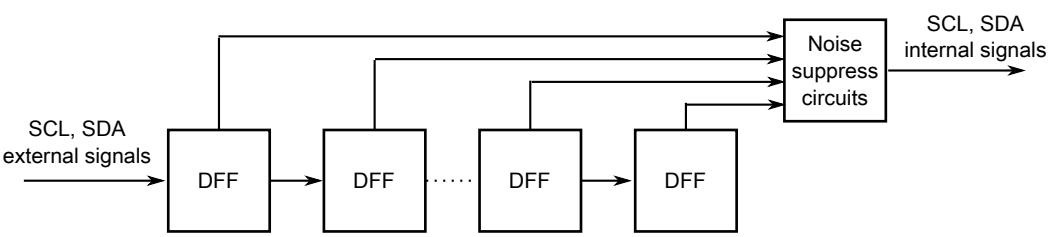


Figure 39-5. Programmable input glitch filter diagram

39.4.8 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).

39.4.9 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.

39.4.10 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.

39.5 Initialization/application information

Module Initialization (Slave)

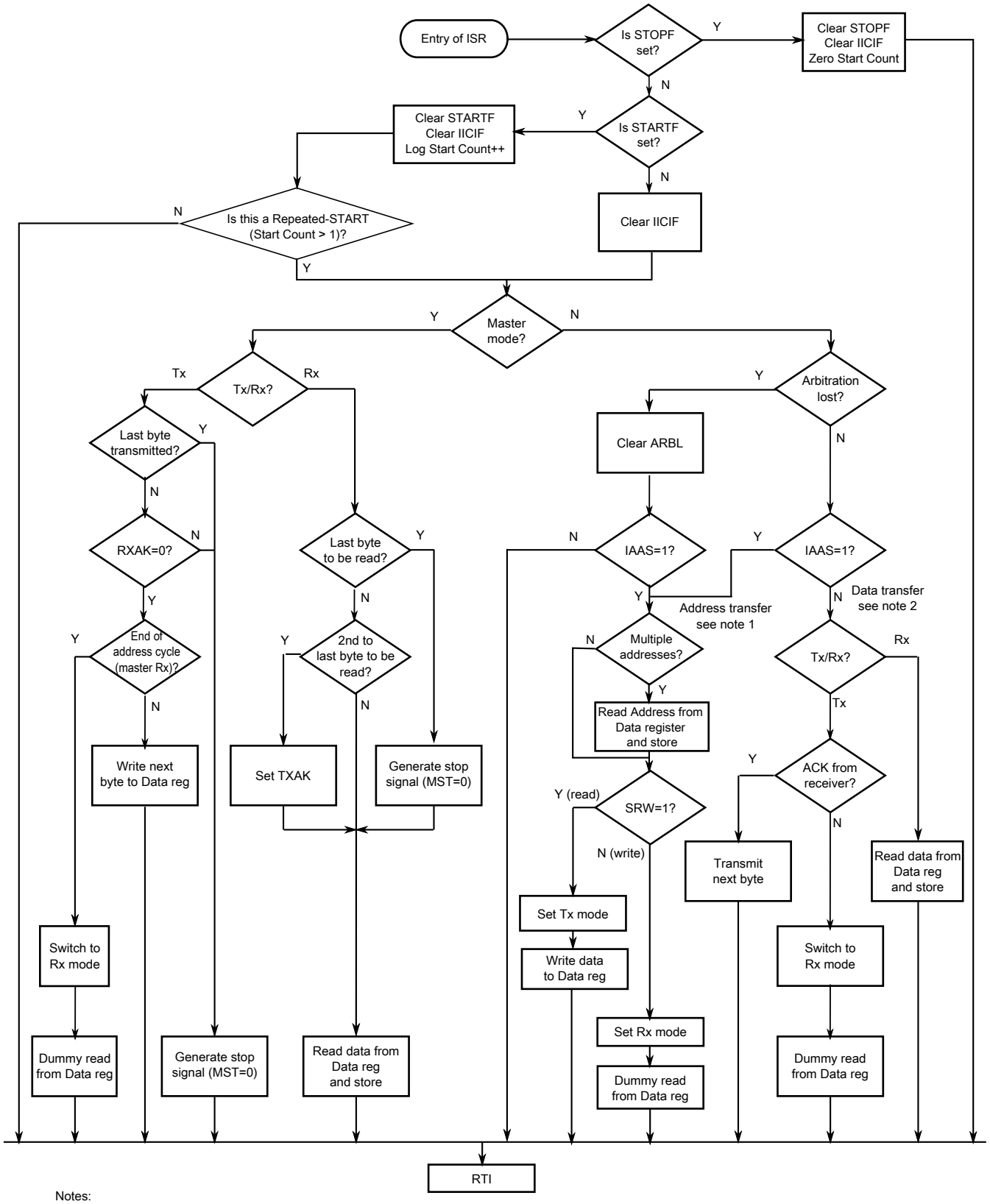
1. Write: Control Register 2
   - to enable or disable general call
   - to select 10-bit or 7-bit addressing mode

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)

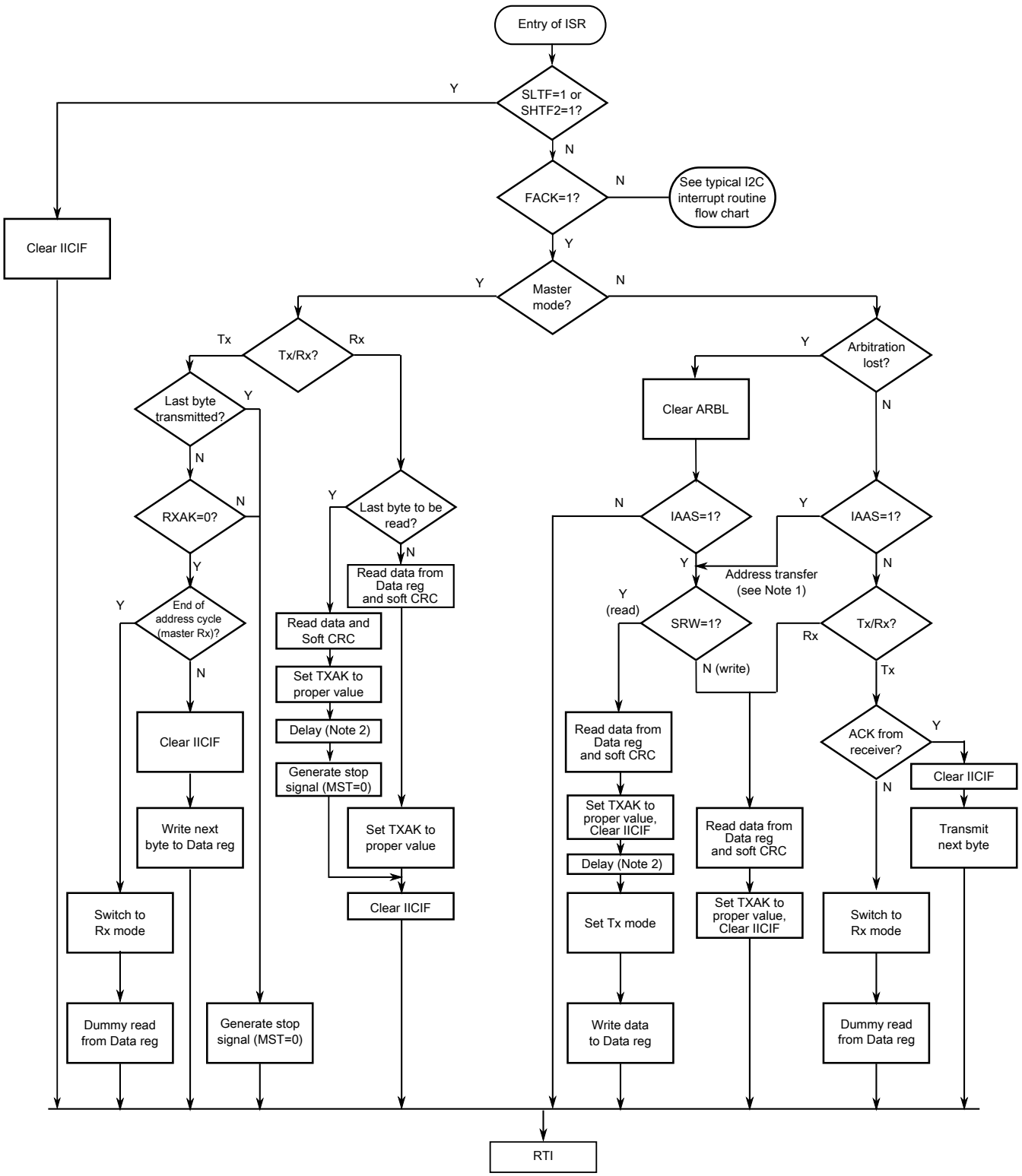
1. Write: Frequency Divider register to set the I2C baud rate (see example in description of ICR)
2. Write: Control Register 1 to enable the I2C module and interrupts
3. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
4. Initialize RAM variables used to achieve the routine shown 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 39-6. 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 39-7. Typical I2C SMBus interrupt routine

Chapter 40
Low Power Universal Asynchronous Receiver/ Transmitter (LPUART)

40.1 Introduction

40.1.1 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

    - Idle line address matching
    - Address match start, address match end
- Optional 13-bit break character generation / 11-bit break character detection
- Configurable idle length detection supporting 1, 2, 4, 8, 16, 32, 64 or 128 idle characters
- 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
- Independent FIFO structure for transmit and receive
    - Separate configurable watermark for receive and transmit requests
    - Option for receiver to assert request after a configurable number of idle characters if receive FIFO is not empty

40.1.2 Modes of operation

40.1.2.1 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.

40.1.3 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

40.1.4 Block diagram

The following figure shows the transmitter portion of the LPUART.

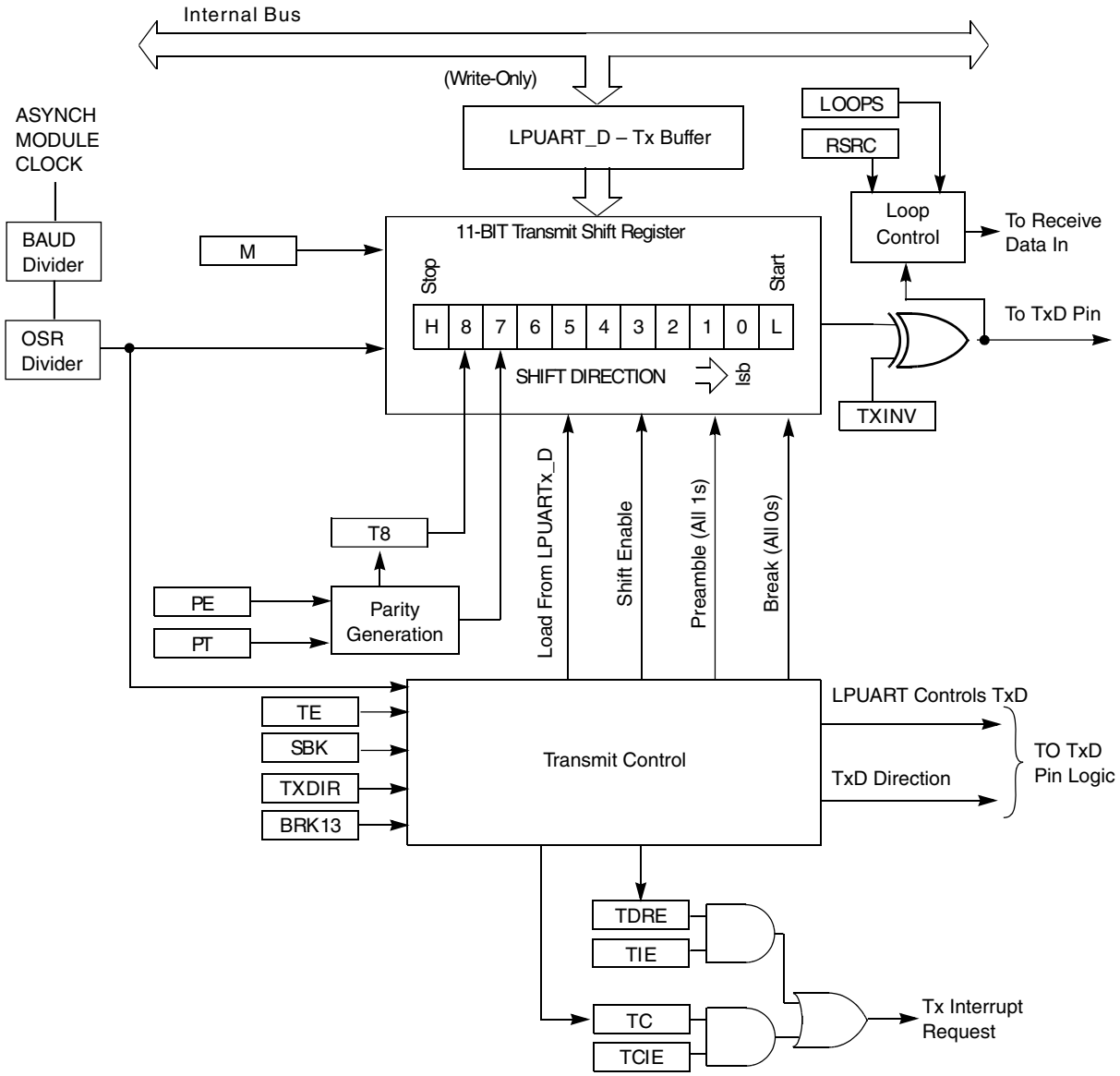


Figure 40-1. LPUART transmitter block diagram

The following figure shows the receiver portion of the LPUART.

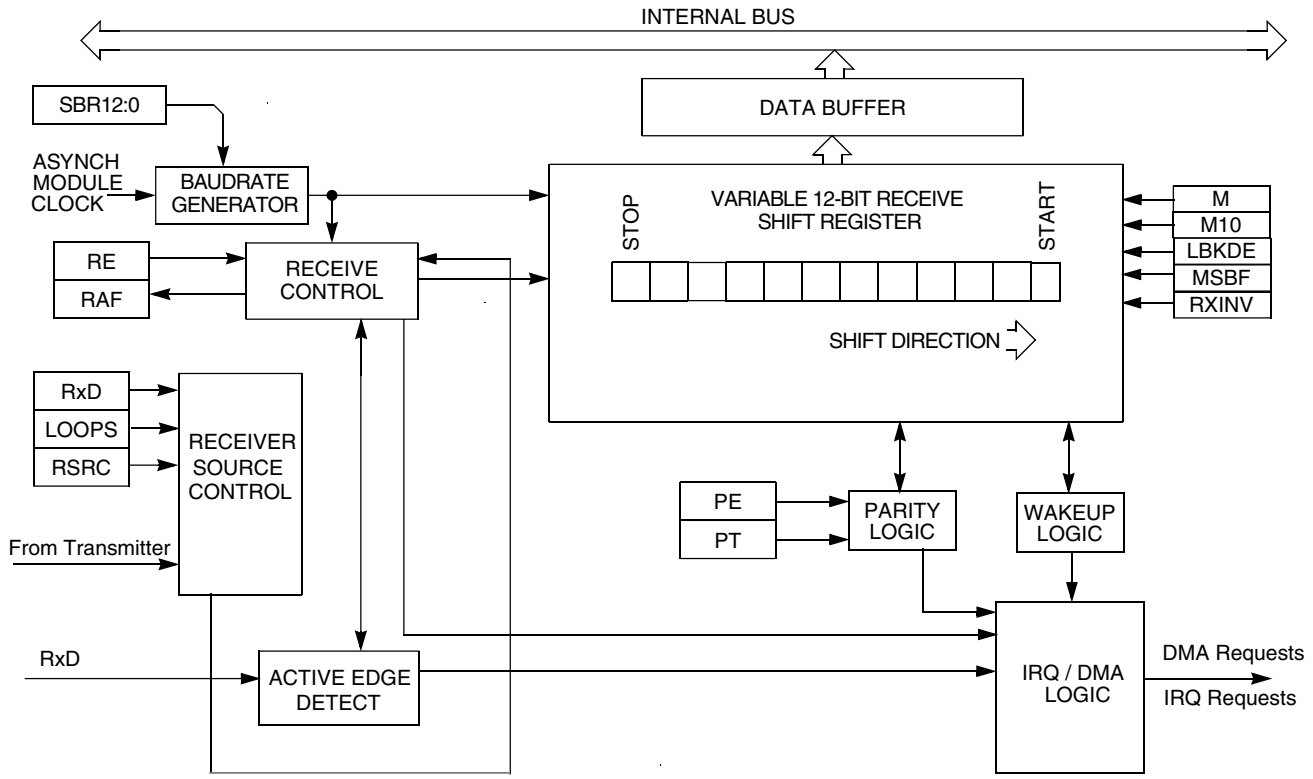


Figure 40-2. LPUART receiver block diagram

40.2 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	40.2.1/1085
4005_4004	LPUART Status Register (LPUART0_STAT)	32	R/W	00C0_0000h	40.2.2/1088
4005_4008	LPUART Control Register (LPUART0_CTRL)	32	R/W	0000_0000h	40.2.3/1092
4005_400C	LPUART Data Register (LPUART0_DATA)	32	R/W	0000_1000h	40.2.4/1097
4005_4010	LPUART Match Address Register (LPUART0_MATCH)	32	R/W	0000_0000h	40.2.5/1099
4005_4014	LPUART Modem IrDA Register (LPUART0_MODIR)	32	R/W	0000_0000h	40.2.6/1099
4005_4018	LPUART FIFO Register (LPUART0_FIFO)	32	R/W	See section	40.2.7/1102
4005_401C	LPUART Watermark Register (LPUART0_WATER)	32	R/W	0000_0000h	40.2.8/1105
4005_5000	LPUART Baud Rate Register (LPUART1_BAUD)	32	R/W	0F00_0004h	40.2.1/1085

Table continues on the next page...

LPUART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_5004	LPUART Status Register (LPUART1_STAT)	32	R/W	00C0_0000h	40.2.2/1088
4005_5008	LPUART Control Register (LPUART1_CTRL)	32	R/W	0000_0000h	40.2.3/1092
4005_500C	LPUART Data Register (LPUART1_DATA)	32	R/W	0000_1000h	40.2.4/1097
4005_5010	LPUART Match Address Register (LPUART1_MATCH)	32	R/W	0000_0000h	40.2.5/1099
4005_5014	LPUART Modem IrDA Register (LPUART1_MODIR)	32	R/W	0000_0000h	40.2.6/1099
4005_5018	LPUART FIFO Register (LPUART1_FIFO)	32	R/W	See section	40.2.7/1102
4005_501C	LPUART Watermark Register (LPUART1_WATER)	32	R/W	0000_0000h	40.2.8/1105

40.2.1 LPUART Baud Rate Register (LPUARTx_BAUD)

Address: Base address + 0h offset

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	MATCFG		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.

Table continues on the next page...

LPUARTx_BAUD field descriptions (continued)

Field	Description
	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.
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 MATCFG	Match Configuration Configures the match addressing mode used. 00 Address Match Wakeup 01 Idle Match Wakeup 10 Match On and Match Off 11 Enables RWU on Data Match and Match On/Off for transmitter CTS input
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

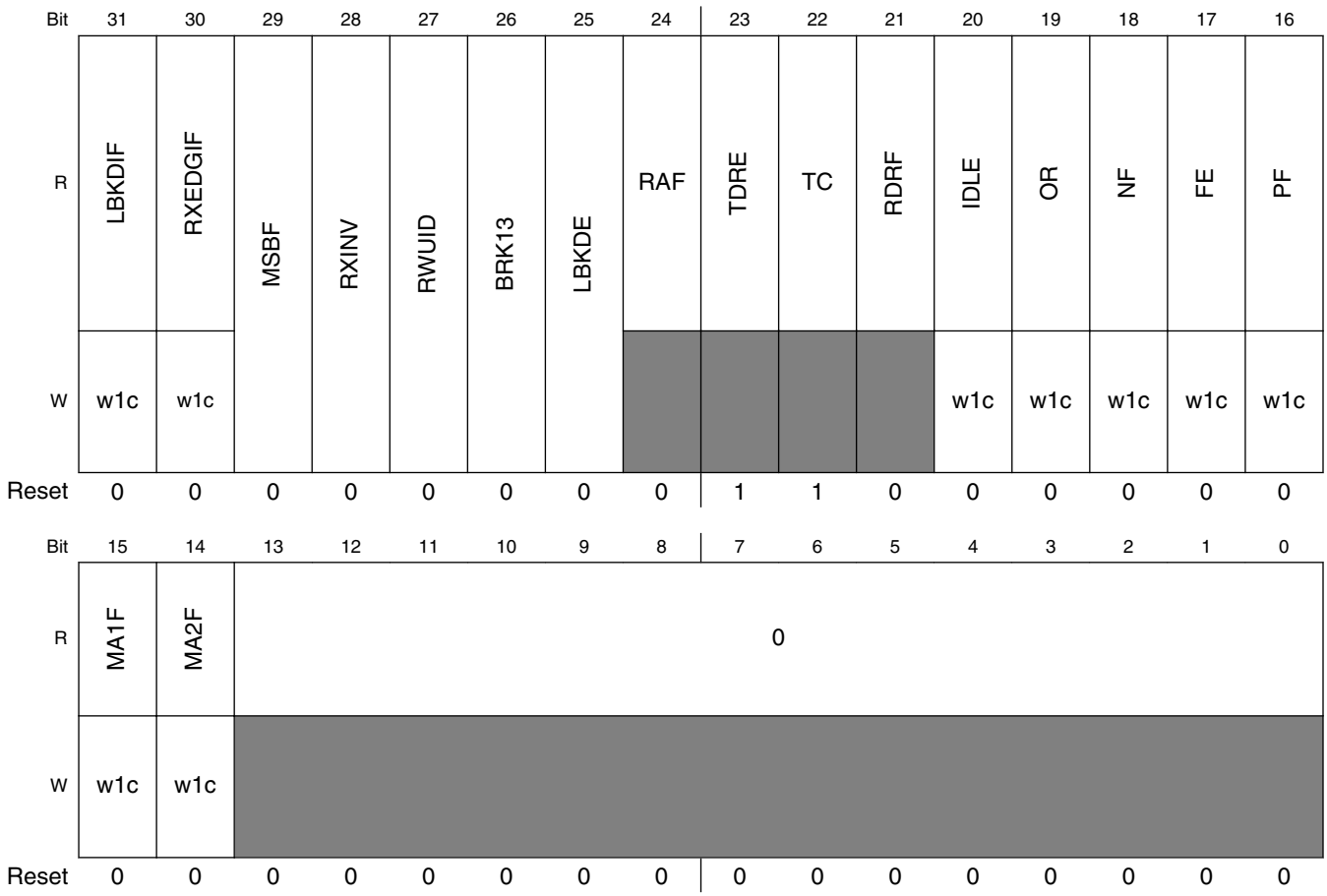
Table continues on the next page...

LPUARTx_BAUD field descriptions (continued)

Field	Description
	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 only be updated when the transmitter and receiver are both disabled (LPUART_CTRL[RE] and LPUART_CTRL[TE] are both 0).

40.2.2 LPUART Status Register (LPUARTx_STAT)

Address: Base address + 4h offset



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 RXEDGIF is set whenever the receiver is enabled and 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

Table continues on the next page...

LPUARTx_STAT field descriptions (continued)

Field	Description
	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. 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.

Table continues on the next page...

LPUARTx_STAT field descriptions (continued)

Field	Description
	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 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.

Table continues on the next page...

LPUARTx_STAT field descriptions (continued)

Field	Description
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.
15 MA1F	Match 1 Flag MA1F is set whenever the next character to be read from LPUART_DATA matches MA1. To clear MA1F, write a logic one to the MA1F. 0 Received data is not equal to MA1 1 Received data is equal to MA1
14 MA2F	Match 2 Flag MA2F is set whenever the next character to be read from LPUART_DATA matches MA2. To clear MA2F, write a logic one to the MA2F. 0 Received data is not equal to MA2 1 Received data is equal to MA2
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

40.2.3 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: Base address + 8h offset

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 W	MA1IE	MA2IE	0			IDLECFG			LOOPS	DOZEEN	RSRC	M	WAKE	ILT	PE	PT
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 MA1IE	Match 1 Interrupt Enable 0 MA1F interrupt disabled 1 MA1F interrupt enabled
14 MA2IE	Match 2 Interrupt Enable 0 MA2F interrupt disabled 1 MA2F interrupt enabled

Table continues on the next page...

LPUARTx_CTRL field descriptions (continued)

Field	Description
13–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 IDLECFG	Idle Configuration Configures the number of idle characters that must be received before the IDLE flag is set. 000 1 idle character 001 2 idle characters 010 4 idle characters 011 8 idle characters 100 16 idle characters 101 32 idle characters 110 64 idle characters 111 128 idle characters
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. 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

Table continues on the next page...

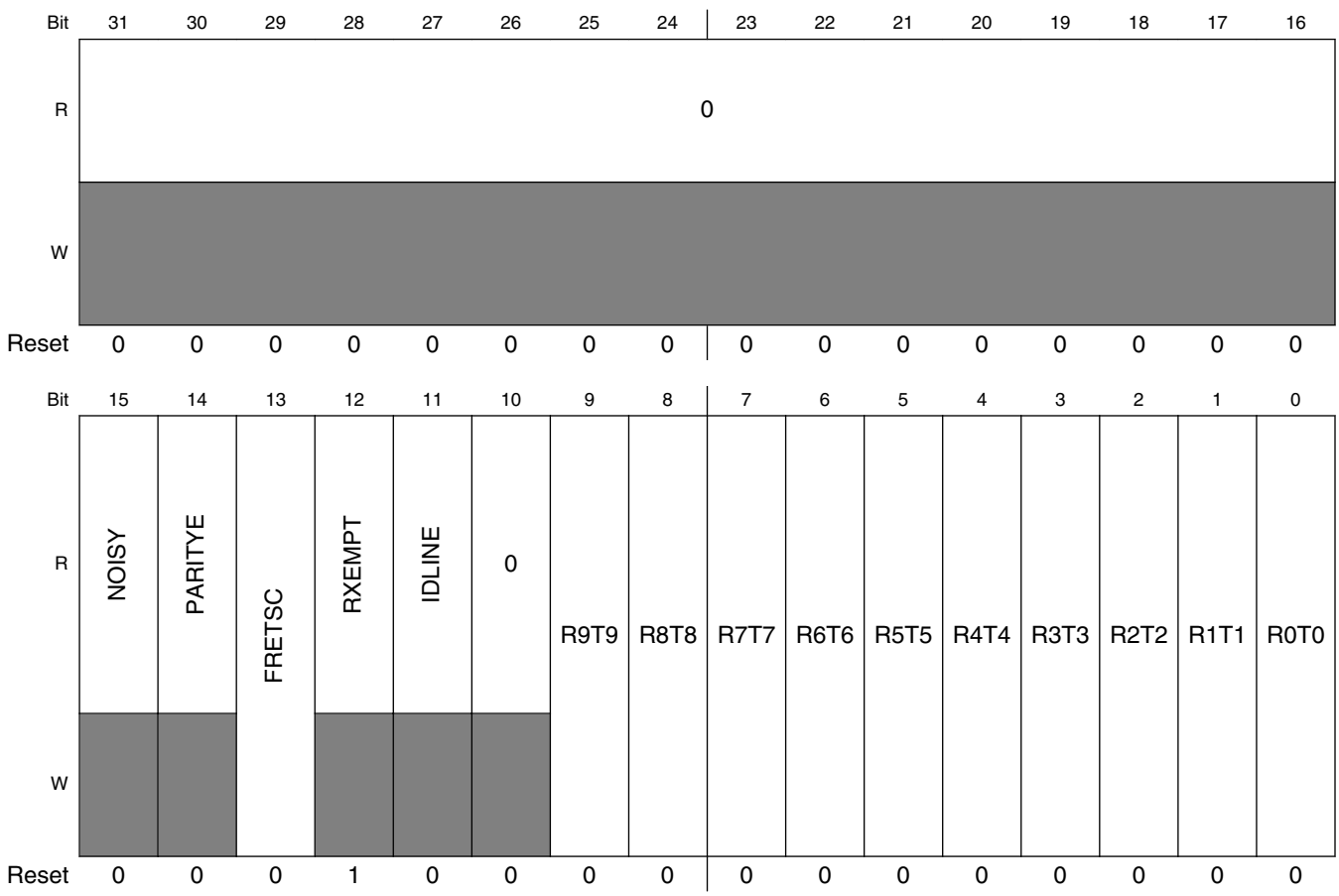
LPUARTx_CTRL field descriptions (continued)

Field	Description
	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. 0 Even parity. 1 Odd parity.

40.2.4 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: Base address + Ch offset



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.

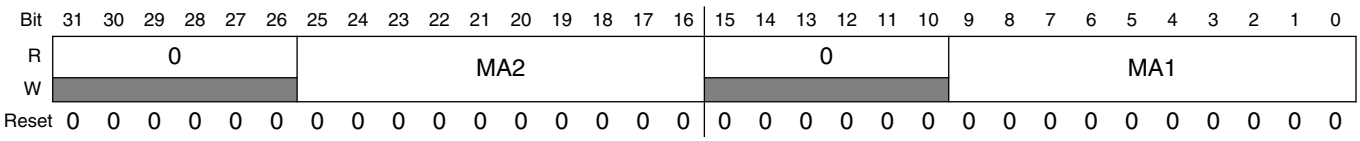
Table continues on the next page...

LPUARTx_DATA field descriptions (continued)

Field	Description
	0 The dataword was received without a parity error. 1 The dataword was received with a parity error.
13 FRETSC	Frame Error / Transmit Special Character For reads, indicates the current received dataword contained in DATA[R9:R0] was received with a frame error. For writes, indicates a break or idle character is to be transmitted instead of the contents in DATA[T9:T0]. T9 is used to indicate a break character when 0 and a idle character when 1, he contents of DATA[T8:T0] should be zero. 0 The dataword was received without a frame error on read, transmit a normal character on write. 1 The dataword was received with a frame error, transmit an idle or break character on transmit.
12 RXEMPT	Receive Buffer Empty Asserts when there is no data in the receive buffer. This field does not take into account data that is in the receive shift register. 0 Receive buffer contains valid data. 1 Receive buffer is empty, data returned on read is not valid.
11 IDLINE	Idle Line Indicates the receiver line was idle before receiving the character in DATA[9:0]. Unlike the IDLE flag, this bit can set for the first character received when the receiver is first enabled. 0 Receiver was not idle before receiving this character. 1 Receiver was idle before receiving this character.
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.

40.2.5 LPUART Match Address Register (LPUARTx_MATCH)

Address: Base address + 10h offset



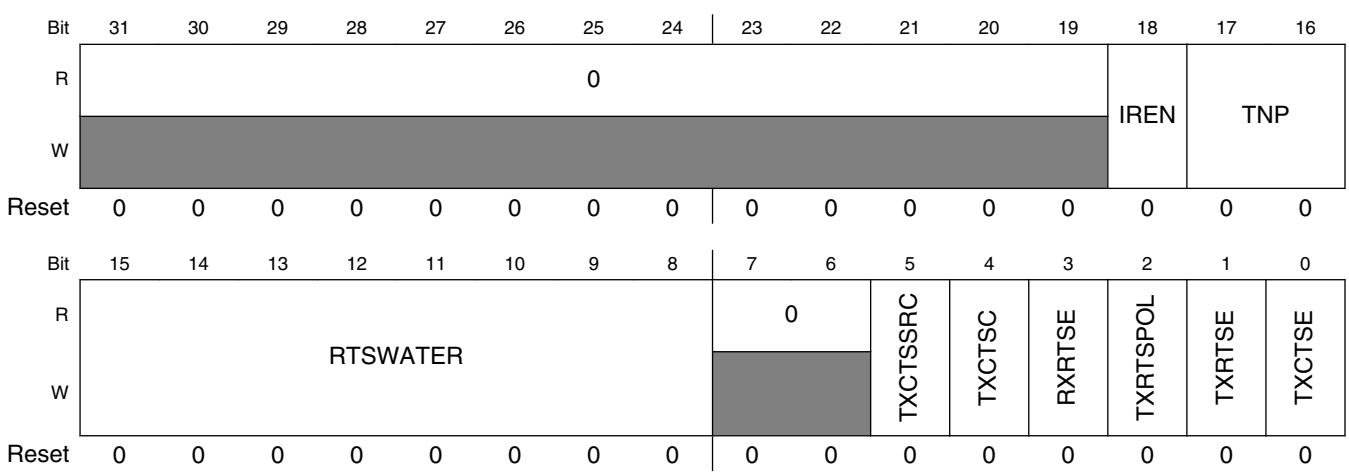
LPUARTx_MATCH field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–16 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.
15–10 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.

40.2.6 LPUART Modem IrDA Register (LPUARTx_MODIR)

The MODEM register controls options for setting the modem configuration.

Address: Base address + 14h offset



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. 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.
15–8 RTSWATER	Receive RTS Configuration Configures the point at which the RX RTS output negates based on the number of additional characters that can be stored in the Receive FIFO. When configured to 0, RTS negates when the the start bit is detetected for the character that will cause the FIFO to become full. 0 RTS asserts when the receiver FIFO is full or receiving a character that causes the FIFO to become full. 1 RTS asserts when the receive FIFO is less than or equal to the RXWATER configuration and negates when the receive FIFO is greater than the RXWATER configuration.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 TXCTSSRC	Transmit CTS Source Configures the source of the CTS input. 0 CTS input is the LPUART_CTS pin. 1 CTS input is the inverted Receiver Match result.
4 TXCTSC	Transmit CTS Configuration Configures if the CTS state is checked at the start of each character or only when the transmitter is idle. 0 CTS input is sampled at the start of each character. 1 CTS input is sampled when the transmitter is idle.
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.

Table continues on the next page...

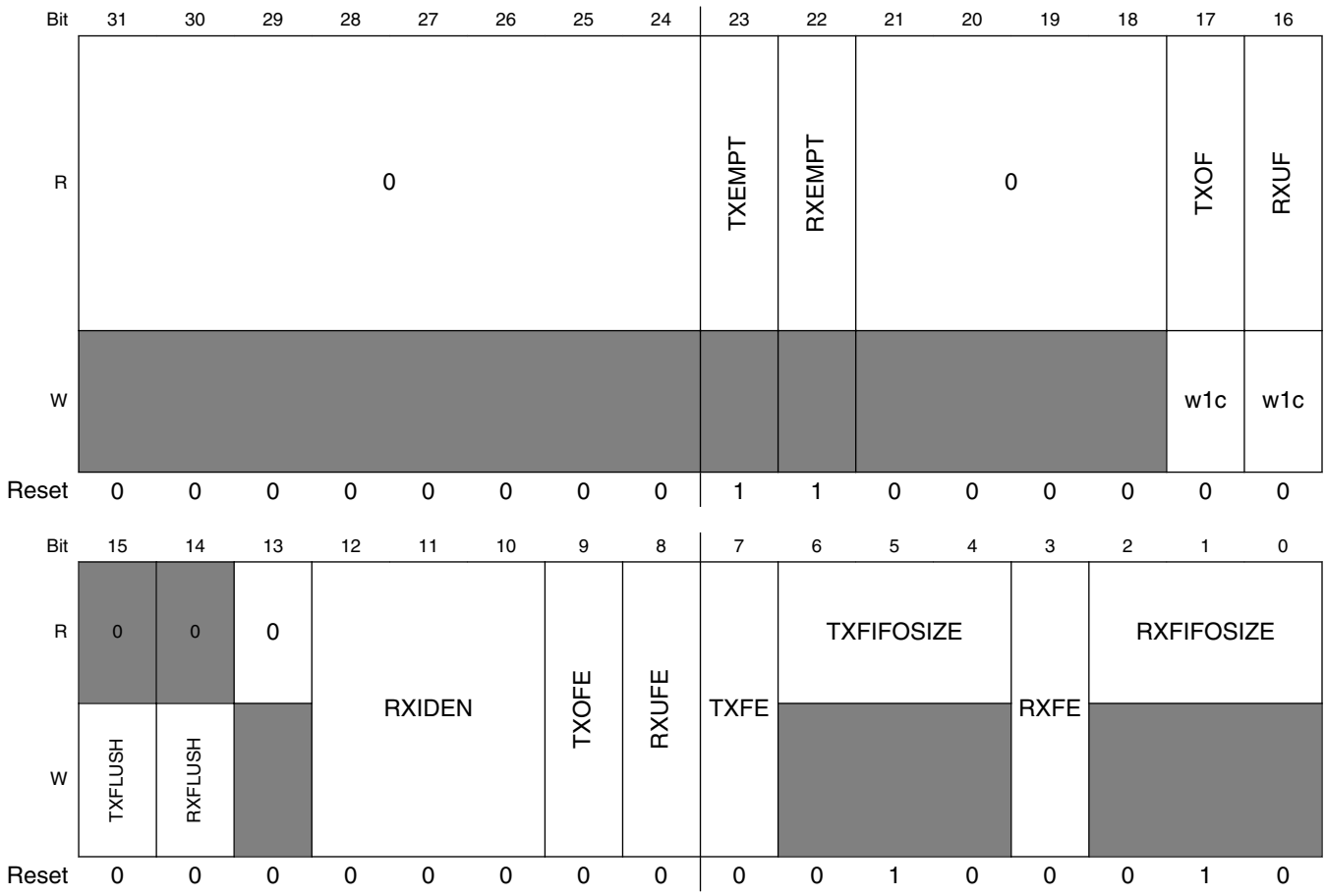
LPUARTx_MODIR field descriptions (continued)

Field	Description
	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.

40.2.7 LPUART FIFO Register (LPUARTx_FIFO)

This register provides the ability for the programmer to turn on and off FIFO functionality. It also provides the size of the FIFO that has been implemented. This register may be read at any time. This register must be written only when CTRL[RE] and CTRL[TE] are cleared/not set and when the data buffer/FIFO is empty.

Address: Base address + 18h offset



LPUARTx_FIFO field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 TXEMPT	Transmit Buffer/FIFO Empty Asserts when there is no data in the Transmit FIFO/buffer. This field does not take into account data that is in the transmit shift register. 0 Transmit buffer is not empty. 1 Transmit buffer is empty.

Table continues on the next page...

LPUARTx_FIFO field descriptions (continued)

Field	Description
22 RXEMPT	Receive Buffer/FIFO Empty Asserts when there is no data in the receive FIFO/Buffer. This field does not take into account data that is in the receive shift register. 0 Receive buffer is not empty. 1 Receive buffer is empty.
21–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 TXOF	Transmitter Buffer Overflow Flag Indicates that more data has been written to the transmit buffer than it can hold. This field will assert regardless of the value of TXOFE. However, an interrupt will be issued to the host only if TXOFE is set. This flag is cleared by writing a 1. 0 No transmit buffer overflow has occurred since the last time the flag was cleared. 1 At least one transmit buffer overflow has occurred since the last time the flag was cleared.
16 RXUF	Receiver Buffer Underflow Flag Indicates that more data has been read from the receive buffer than was present. This field will assert regardless of the value of RXUFE. However, an interrupt will be issued to the host only if RXUFE is set. This flag is cleared by writing a 1. 0 No receive buffer underflow has occurred since the last time the flag was cleared. 1 At least one receive buffer underflow has occurred since the last time the flag was cleared.
15 TXFLUSH	Transmit FIFO/Buffer Flush Writing to this field causes all data that is stored in the transmit FIFO/buffer to be flushed. This does not affect data that is in the transmit shift register. 0 No flush operation occurs. 1 All data in the transmit FIFO/Buffer is cleared out.
14 RXFLUSH	Receive FIFO/Buffer Flush Writing to this field causes all data that is stored in the receive FIFO/buffer to be flushed. This does not affect data that is in the receive shift register. 0 No flush operation occurs. 1 All data in the receive FIFO/buffer is cleared out.
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–10 RXIDEN	Receiver Idle Empty Enable When set, enables the assertion of RDRF when the receiver is idle for a number of idle characters and the FIFO is not empty. 000 Disable RDRF assertion due to partially filled FIFO when receiver is idle. 001 Enable RDRF assertion due to partially filled FIFO when receiver is idle for 1 character. 010 Enable RDRF assertion due to partially filled FIFO when receiver is idle for 2 characters. 011 Enable RDRF assertion due to partially filled FIFO when receiver is idle for 4 characters. 100 Enable RDRF assertion due to partially filled FIFO when receiver is idle for 8 characters. 101 Enable RDRF assertion due to partially filled FIFO when receiver is idle for 16 characters.

Table continues on the next page...

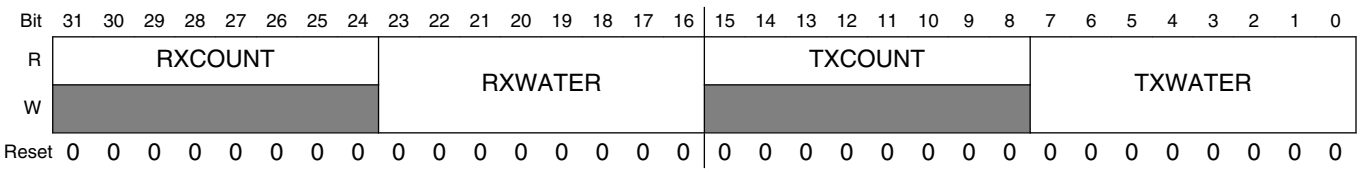
LPUARTx_FIFO field descriptions (continued)

Field	Description
	110 Enable RDRF assertion due to partially filled FIFO when receiver is idle for 32 characters. 111 Enable RDRF assertion due to partially filled FIFO when receiver is idle for 64 characters.
9 TXOFE	Transmit FIFO Overflow Interrupt Enable When this field is set, the TXOF flag generates an interrupt to the host. 0 TXOF flag does not generate an interrupt to the host. 1 TXOF flag generates an interrupt to the host.
8 RXUFE	Receive FIFO Underflow Interrupt Enable When this field is set, the RXUF flag generates an interrupt to the host. 0 RXUF flag does not generate an interrupt to the host. 1 RXUF flag generates an interrupt to the host.
7 TXFE	Transmit FIFO Enable When this field is set, the built in FIFO structure for the transmit buffer is enabled. The size of the FIFO structure is indicated by TXFIFOSIZE. If this field is not set, the transmit buffer operates as a FIFO of depth one dataword regardless of the value in TXFIFOSIZE. Both CTRL[TE] and CTRL[RE] must be cleared prior to changing this field. 0 Transmit FIFO is not enabled. Buffer is depth 1. (Legacy support). 1 Transmit FIFO is enabled. Buffer is depth indicated by TXFIFOSIZE.
6–4 TXFIFOSIZE	Transmit FIFO. Buffer Depth The maximum number of transmit datawords that can be stored in the transmit buffer. This field is read only. 010 Transmit FIFO/Buffer depth = 4 datawords.
3 RXFE	Receive FIFO Enable When this field is set, the built in FIFO structure for the receive buffer is enabled. The size of the FIFO structure is indicated by the RXFIFOSIZE field. If this field is not set, the receive buffer operates as a FIFO of depth one dataword regardless of the value in RXFIFOSIZE. Both CTRL[TE] and CTRL[RE] must be cleared prior to changing this field. 0 Receive FIFO is not enabled. Buffer is depth 1. (Legacy support) 1 Receive FIFO is enabled. Buffer is depth indicted by RXFIFOSIZE.
RXFIFOSIZE	Receive FIFO. Buffer Depth The maximum number of receive datawords that can be stored in the receive buffer before an overrun occurs. This field is read only. 010 Receive FIFO/Buffer depth = 4 datawords.

40.2.8 LPUART Watermark Register (LPUARTx_WATER)

This register provides the ability to set a programmable threshold for notification of needing additional transmit data. This register may be read at any time but must be written only when CTRL[TE] is not set.

Address: Base address + 1Ch offset



LPUARTx_WATER field descriptions

Field	Description
31–24 RXCOUNT	Receive Counter The value in this register indicates the number of datawords that are in the receive FIFO/buffer. If a dataword is being received, that is, in the receive shift register, it is not included in the count. This value may be used in conjunction with FIFO[RXFIFOSIZE] to calculate how much room is left in the receive FIFO/buffer.
23–16 RXWATER	Receive Watermark When the number of datawords in the receive FIFO/buffer is greater than the value in this register field, an interrupt or a DMA request is generated. For proper operation, the value in RXWATER must be set to be less than the receive FIFO/buffer size as indicated by FIFO[RXFIFOSIZE] and FIFO[RXFE] and must be greater than 0.
15–8 TXCOUNT	Transmit Counter The value in this register indicates the number of datawords that are in the transmit FIFO/buffer. If a dataword is being transmitted, that is, in the transmit shift register, it is not included in the count. This value may be used in conjunction with FIFO[TXFIFOSIZE] to calculate how much room is left in the transmit FIFO/buffer.
TXWATER	Transmit Watermark When the number of datawords in the transmit FIFO/buffer is equal to or less than the value in this register field, an interrupt or a DMA request is generated. For proper operation, the value in TXWATER must be set to be less than the size of the transmit buffer/FIFO size as indicated by FIFO[TXFIFOSIZE] and FIFO[TXFE].

40.3 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.

40.3.1 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.

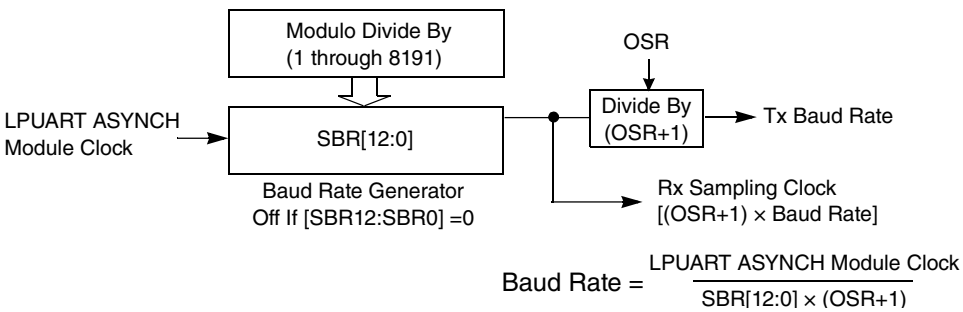


Figure 40-3. LPUART baud rate generation

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.

40.3.2 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.

40.3.2.1 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 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 40-1. 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

40.3.2.2 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.

40.3.2.3 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.

40.3.2.4 Transceiver driver enable using LPUART_RTS

RS-485 is a multiple drop communication protocol in which the LPUART transceiver's driver is 3-stated unless the UART is driving. The LPUART_RTS signal can be used by the transmitter to enable the driver of a transceiver. The polarity of LPUART_RTS can be matched to the polarity of the transceiver's driver enable signal.

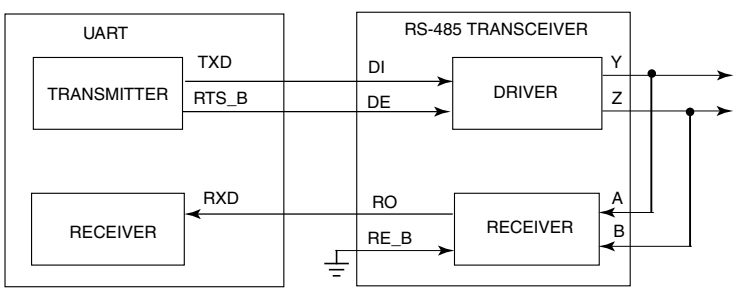


Figure 40-4. Transceiver driver enable using LPUART_RTS

In the figure, the receiver enable signal is asserted. Another option for this connection is to connect LPUART_RTS to both DE and RE_B. The transceiver's receiver is disabled while driving. A pullup can pull LPUART_RX to a non-floating value during this time. This option can be refined further by operating the LPUART in single wire mode, freeing the LPUART_RX pin for other uses.

40.3.3 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.

40.3.3.1 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.

40.3.3.2 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 40-2. Receiver Wakeup Options

RWU	MA1 | MA2	MATCFG	WAKE:RWUID	Receiver Wakeup
0	0	X	X	Normal operation
1	0	00	00	Receiver wakeup on idle line, IDLE flag not set
1	0	00	01	Receiver wakeup on idle line, IDLE flag set
1	0	00	10	Receiver wakeup on address mark
1	1	11	X0	Receiver wakeup on data match
0	1	00	X0	Address mark address match, IDLE flag not set for discarded characters
0	1	00	X1	Address mark address match, IDLE flag set for discarded characters
0	1	01	X0	Idle line address match
0	1	10	X0	Address match on and address match off, IDLE flag not set for discarded characters
0	1	10	X1	Address match on and address match off, IDLE flag set for discarded characters

40.3.3.2.1 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.

40.3.3.2.2 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.

40.3.3.2.3 Data match wakeup

When LPUART_CTRL[RWU] is set and LPUART_BAUD[MATCFG] equals 11, the receiver is configured for data match wakeup. In this mode, LPUART_CTRL[RWU] is cleared automatically when the receiver detects a character that matches MATCH[MA1] field when BAUD[MAEN1] is set, or that matches MATCH[MA2] when BAUD[MAEN2] is set.

40.3.3.2.4 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.

40.3.3.2.5 Idle Match operation

Idle match operation is enabled when the LPUART_BAUD[MAEN1] or LPUART_BAUD[MAEN2] bit is set and LPUART_BAUD[MATCFG] is equal to 01. In this function, the first character received by the LPUART_RX pin after an idle line condition is considered an address and is compared with the associated MA1 or MA2 register. The character is only transferred to the receive buffer, and LPUART_STAT[RDRF] is set, if the comparison matches. All subsequent characters are considered to be data associated with the address and are transferred to the receive data buffer until the next idle line condition is detected. If no address match occurs then no transfer is made to the receive data buffer, and all following frames until the next idle condition 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.

Idle match operation functions in the same way for both MA1 and MA2 registers.

- If only one of LPUART_BAUD[MAEN1] and LPUART_BAUD[MAEN2] is asserted, the first character after an idle line 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, the first character after an idle line is compared with both match registers and data is transferred only on a match with either register.

40.3.3.2.6 Match On Match Off operation

Match on, match off operation is enabled when both LPUART_BAUD[MAEN1] and LPUART_BAUD[MAEN2] are set and LPUART_BAUD[MATCFG] is equal to 10. In this function, a character received by the LPUART_RX pin that matches MATCH[MA1] is received and transferred to the receive buffer, and LPUART_STAT[RDRF] is set. All subsequent characters are considered to be data and are also transferred to the receive

data buffer, until a character is received that matches MATCH[MA2] register. The character that matches MATCH[MA2] and all following characters are discarded, this continues until another character that matches MATCH[MA1] is received. 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.

NOTE

Match on, match off operation requires both LPUART_BAUD[MAEN1] and LPUART_BAUD[MAEN2] to be asserted.

40.3.3.3 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.

40.3.3.4 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.

40.3.3.4.1 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.

40.3.3.4.2 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.

40.3.3.4.3 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.

40.3.3.4.4 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.

40.3.4 Additional LPUART functions

The following sections describe additional LPUART functions.

40.3.4.1 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.

40.3.4.2 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.

The number of idle characters that must be received before an idle line condition is detected can also be configured using the CTRL[IDLECFG] field. This field configures the number of idle characters that must be received before the STAT[IDLE] flag is set, the STAT[RAF] flag is cleared and the DATA[IDLINE] flag is set with the next received character.

Idle-line wakeup and idle match operation are also affected by the CTRL[IDLECFG] field. When address match or match on/off operation is enabled, setting the STAT[RWUID] bit will cause any discarded characters to be treated as if they were idle characters.

40.3.4.3 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.

40.3.4.4 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.

40.3.5 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 LPUART. 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.

40.3.5.1 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.

40.3.5.2 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.

40.3.6 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.

If the received character matches the contents of MATCH[MA1] and/or MATCH[MA2] then the LPUART_STAT[MA1F] and/or LPUART_STAT[MA2F] flags are set at the same time that LPUART_STAT[RDRF] is set.

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 41
Carrier Modulator Transmitter (CMT)

41.1 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.

41.2 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

41.3 Block diagram

The following figure presents the block diagram of the CMT module.

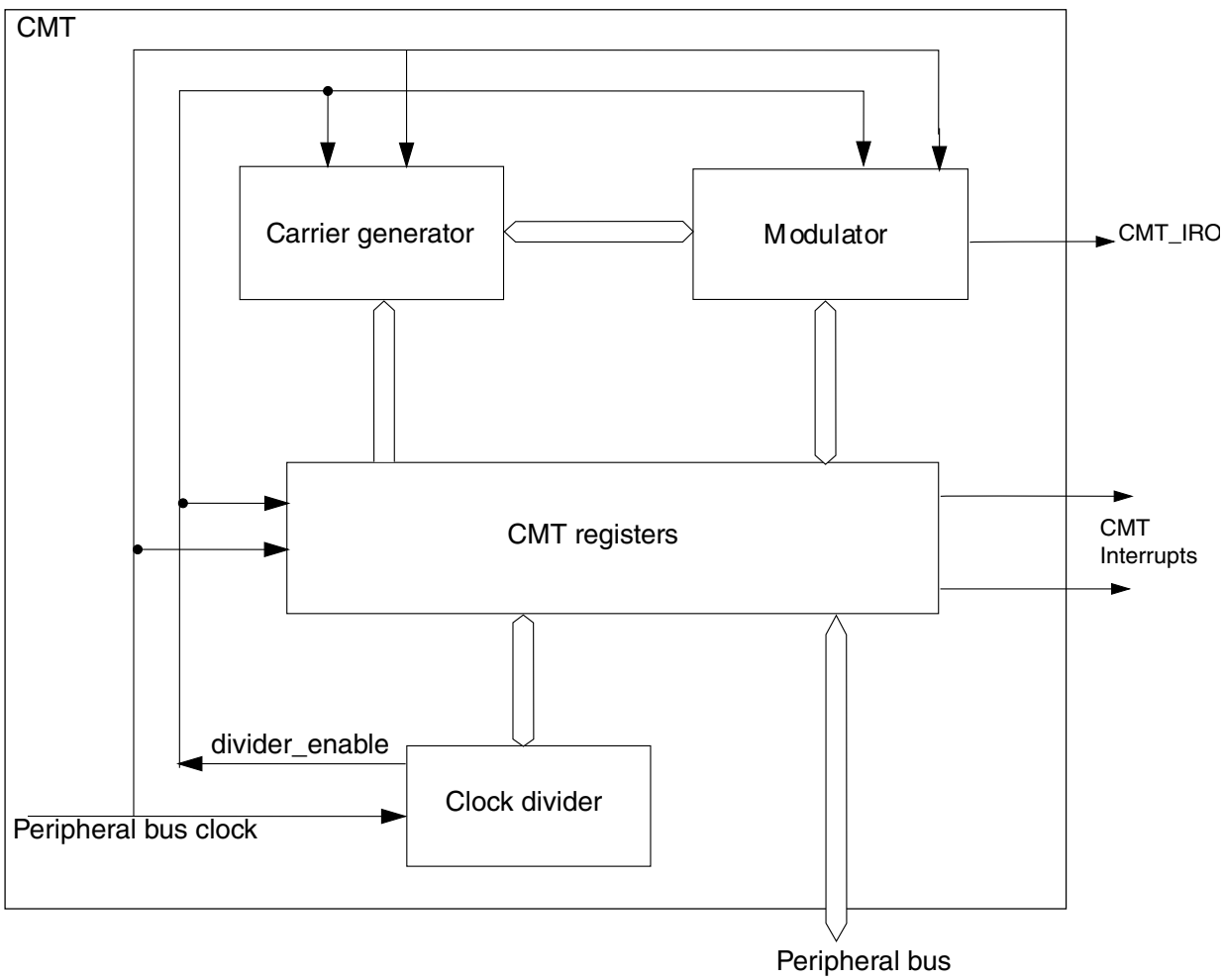


Figure 41-1. CMT module block diagram

41.4 Modes of operation

The following table describes the operation of the CMT module operates in various modes.

Table 41-1. 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 41-2. 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.

41.4.1 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.

41.4.2 Stop mode operation

This section describes the CMT Stop mode operations.

41.4.2.1 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.

41.4.2.2 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.

41.5 CMT external signal descriptions

The following table shows the description of the external signal.

Table 41-3. CMT signal description

Signal	Description	I/O
CMT_IRO	Infrared Output	O

41.5.1 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 41-4 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 41-4. CMT_IRO signal delay calculation

Condition	Delay (bus clock cycles)
MSC[CMTDIV] = 0	PPS[PPSDIV] + 2
MSC[CMTDIV] > 0	(PPS[PPSDIV] *2) + 3

41.6 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	41.6.1/1126
4006_2001	CMT Carrier Generator Low Data Register 1 (CMT_CGL1)	8	R/W	Undefined	41.6.2/1127
4006_2002	CMT Carrier Generator High Data Register 2 (CMT_CGH2)	8	R/W	Undefined	41.6.3/1127
4006_2003	CMT Carrier Generator Low Data Register 2 (CMT_CGL2)	8	R/W	Undefined	41.6.4/1128
4006_2004	CMT Output Control Register (CMT_OC)	8	R/W	00h	41.6.5/1128
4006_2005	CMT Modulator Status and Control Register (CMT_MSC)	8	R/W	00h	41.6.6/1129
4006_2006	CMT Modulator Data Register Mark High (CMT_CMD1)	8	R/W	Undefined	41.6.7/1131
4006_2007	CMT Modulator Data Register Mark Low (CMT_CMD2)	8	R/W	Undefined	41.6.8/1132
4006_2008	CMT Modulator Data Register Space High (CMT_CMD3)	8	R/W	Undefined	41.6.9/1132
4006_2009	CMT Modulator Data Register Space Low (CMT_CMD4)	8	R/W	Undefined	41.6.10/ 1133
4006_200A	CMT Primary Prescaler Register (CMT_PPS)	8	R/W	00h	41.6.11/ 1133
4006_200B	CMT Direct Memory Access Register (CMT_DMA)	8	R/W	00h	41.6.12/ 1134

41.6.1 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.

41.6.2 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.

41.6.3 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.

41.6.4 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.

41.6.5 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.

41.6.6 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

41.6.7 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.

41.6.8 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.

41.6.9 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.

41.6.10 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.

41.6.11 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

41.6.12 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.

41.7 Functional description

The CMT module primarily consists of clock divider, carrier generator, and modulator.

41.7.1 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.

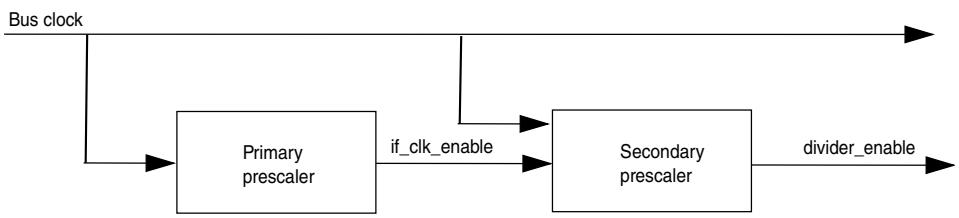


Figure 41-2. Clock divider block diagram

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.

41.7.2 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 41-5. 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.

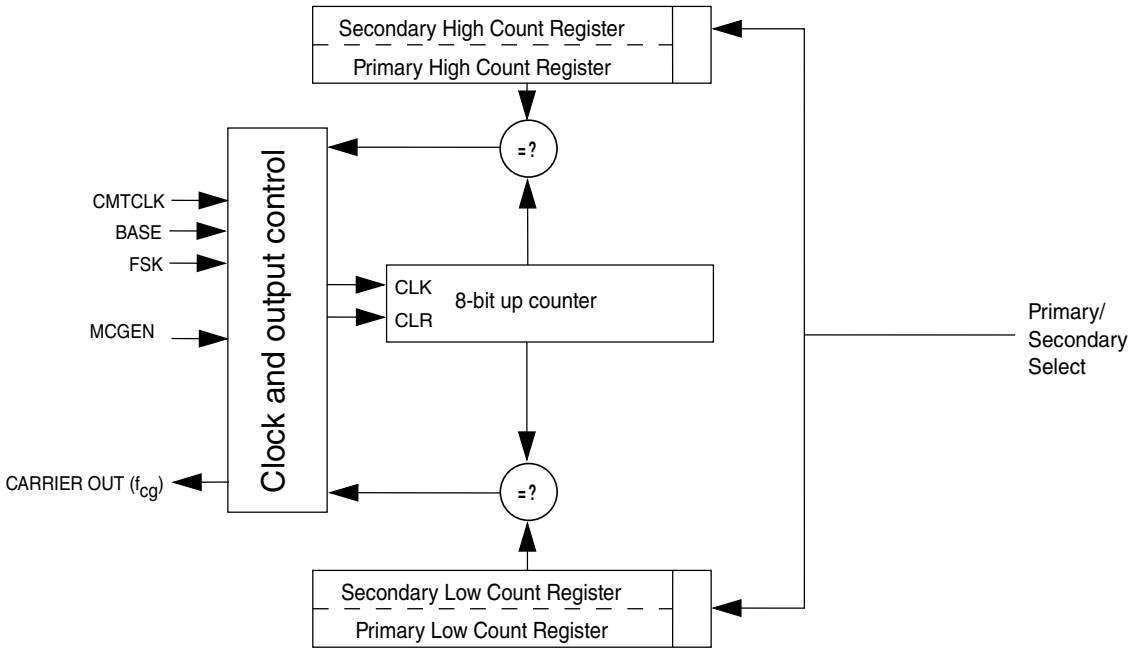


Figure 41-3. Carrier generator block diagram

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.

$$\text{DutyCycle} = \frac{\text{Highcount}}{\text{Highcount+Lowcount}}$$

41.7.3 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 41-6. 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.

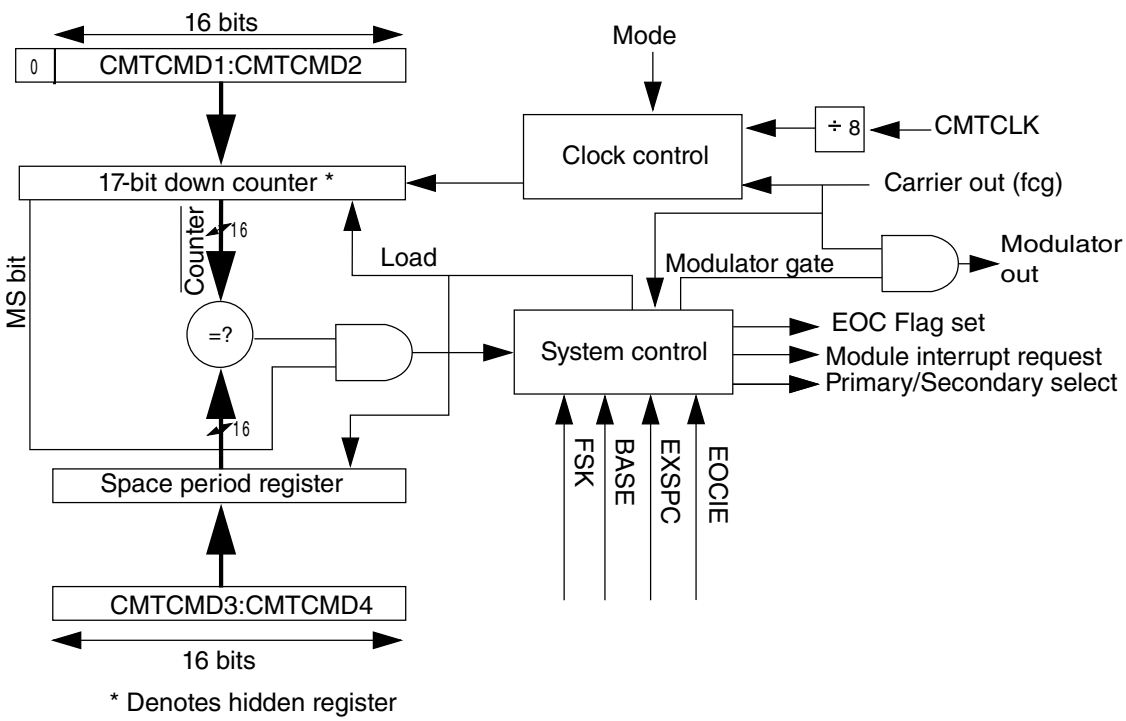


Figure 41-4. Modulator block diagram

41.7.3.1 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 41-5 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) ÷ (f_{CMTCLK} ÷ 8)

t_{space} = CMD3:CMD4 ÷ (f_{CMTCLK} ÷ 8)

where CMD1:CMD2 and CMD3:CMD4 are the decimal values of the concatenated registers.

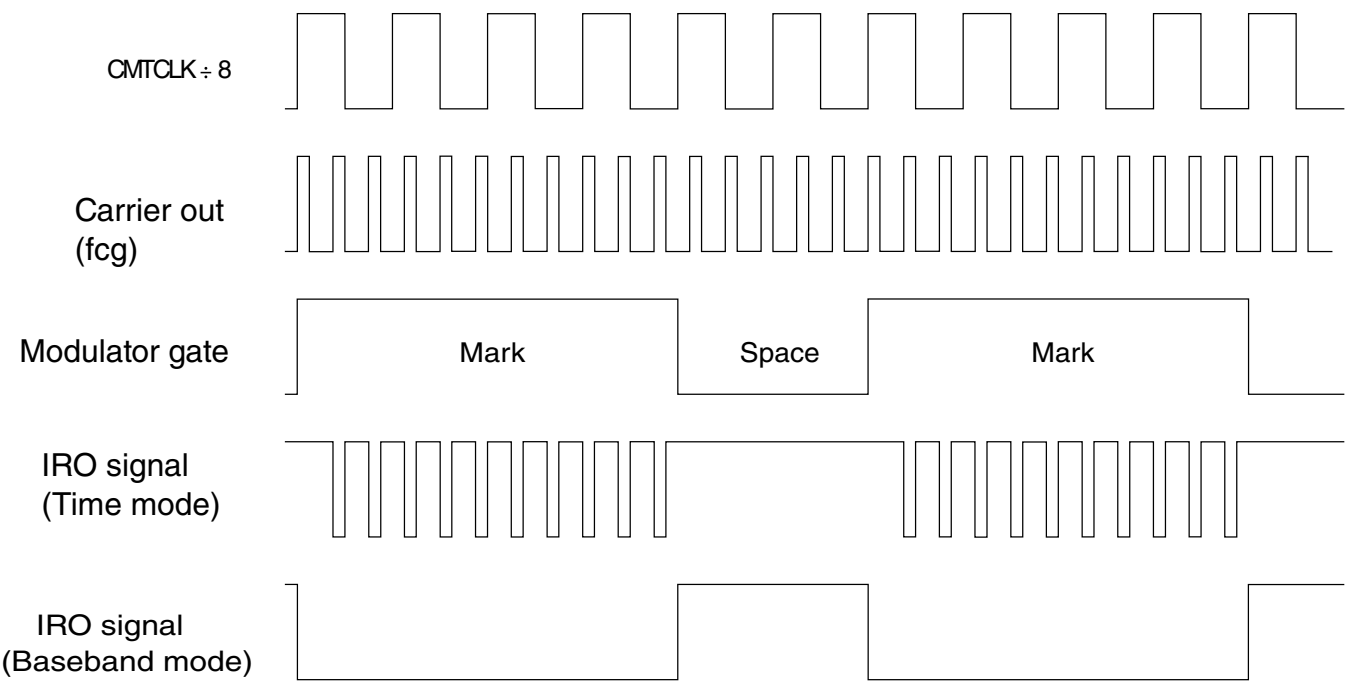


Figure 41-5. Example: CMT output in Time and Baseband modes with OC[CMTPOL]=0

41.7.3.2 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 41-5 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 41-5 and Figure 41-6 are for the purpose of conceptual illustration and are not meant to represent precise timing relationships between the signals shown.

41.7.3.3 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} = (CMD1:CMD2 + 1) ÷ f_{cg}

t_{space} = (CMD3:CMD4) ÷ f_{cg}

Where f_{cg} is the frequency output from the carrier generator. The example in Figure 41-6 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

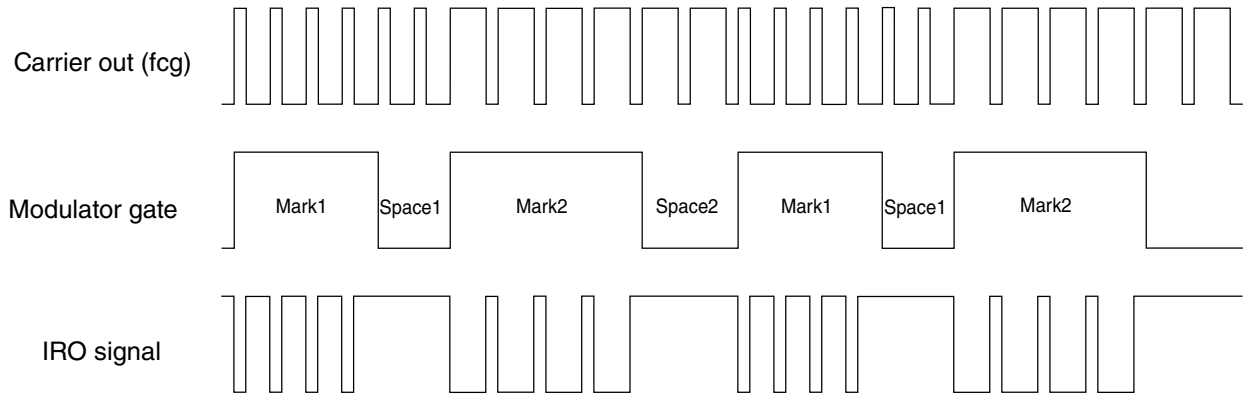


Figure 41-6. Example: CMT output in FSK mode

41.7.4 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.

41.7.4.1 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 41-7.

Note

The extended space enable feature can be used to emulate a zero mark event.

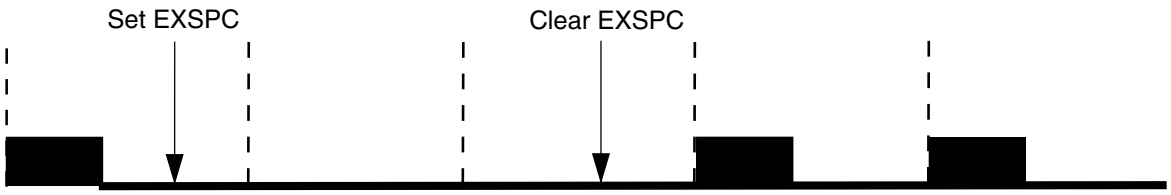


Figure 41-7. Extended space operation

41.7.4.2 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.

41.8 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 41-7. 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 41-8. 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 42
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.

42.1 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.

42.1.1 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.

42.1.2 Modes of operation

The following table depicts different modes of operation and the behavior of the GPIO module in these modes.

Table 42-1. 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.

42.1.3 GPIO signal descriptions

Table 42-2. 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.

42.1.3.1 Detailed signal description

Table 42-3. GPIO interface-detailed signal descriptions

Signal	I/O	Description	
PORTA31–PORTA0 PORTB31–PORTB0 PORTC31–PORTC0	I/O	General-purpose input/output	
		State meaning	Asserted: The pin is logic 1. Deasserted: The pin is logic 0.
		Timing	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.

Table 42-3. 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.

42.2 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 chip-specific GPIO information 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	42.2.1/1150
400F_F004	Port Set Output Register (GPIOA_PSOR)	32	W (always reads 0)	0000_0000h	42.2.2/1151
400F_F008	Port Clear Output Register (GPIOA_PCOR)	32	W (always reads 0)	0000_0000h	42.2.3/1151
400F_F00C	Port Toggle Output Register (GPIOA_PTOR)	32	W (always reads 0)	0000_0000h	42.2.4/1152
400F_F010	Port Data Input Register (GPIOA_PDIR)	32	R	0000_0000h	42.2.5/1152
400F_F014	Port Data Direction Register (GPIOA_PDDR)	32	R/W	0000_0000h	42.2.6/1153

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	42.2.1/1150
400F_F044	Port Set Output Register (GPIOB_PSOR)	32	W (always reads 0)	0000_0000h	42.2.2/1151
400F_F048	Port Clear Output Register (GPIOB_PCOR)	32	W (always reads 0)	0000_0000h	42.2.3/1151
400F_F04C	Port Toggle Output Register (GPIOB_PTOR)	32	W (always reads 0)	0000_0000h	42.2.4/1152
400F_F050	Port Data Input Register (GPIOB_PDIR)	32	R	0000_0000h	42.2.5/1152
400F_F054	Port Data Direction Register (GPIOB_PDDR)	32	R/W	0000_0000h	42.2.6/1153
400F_F080	Port Data Output Register (GPIOC_PDOR)	32	R/W	0000_0000h	42.2.1/1150
400F_F084	Port Set Output Register (GPIOC_PSOR)	32	W (always reads 0)	0000_0000h	42.2.2/1151
400F_F088	Port Clear Output Register (GPIOC_PCOR)	32	W (always reads 0)	0000_0000h	42.2.3/1151
400F_F08C	Port Toggle Output Register (GPIOC_PTOR)	32	W (always reads 0)	0000_0000h	42.2.4/1152
400F_F090	Port Data Input Register (GPIOC_PDIR)	32	R	0000_0000h	42.2.5/1152
400F_F094	Port Data Direction Register (GPIOC_PDDR)	32	R/W	0000_0000h	42.2.6/1153

42.2.1 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.

42.2.2 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.

42.2.3 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

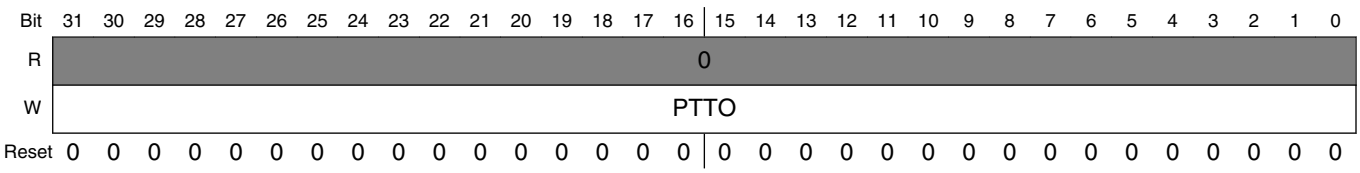
Field	Description
PTCO	Port Clear Output

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.

42.2.4 Port Toggle Output Register (GPIOx_PTOR)

Address: Base address + Ch offset



GPIOx_PTOR field descriptions

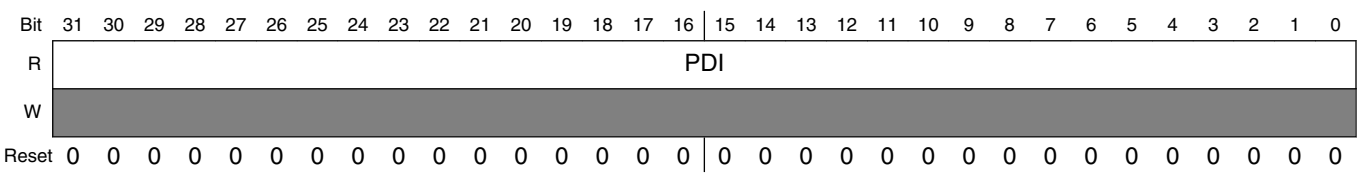
Field	Description
PTTO	Port Toggle Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to the inverse of its existing logic state.

42.2.5 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



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.

42.2.6 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.

42.3 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	42.3.1/1155
F800_0004	Port Set Output Register (FGPIOA_PSOR)	32	W (always reads 0)	0000_0000h	42.3.2/1155
F800_0008	Port Clear Output Register (FGPIOA_PCOR)	32	W (always reads 0)	0000_0000h	42.3.3/1156
F800_000C	Port Toggle Output Register (FGPIOA_PTOR)	32	W (always reads 0)	0000_0000h	42.3.4/1156
F800_0010	Port Data Input Register (FGPIOA_PDIR)	32	R	0000_0000h	42.3.5/1157
F800_0014	Port Data Direction Register (FGPIOA_PDDR)	32	R/W	0000_0000h	42.3.6/1157
F800_0040	Port Data Output Register (FGPIOB_PDOR)	32	R/W	0000_0000h	42.3.1/1155
F800_0044	Port Set Output Register (FGPIOB_PSOR)	32	W (always reads 0)	0000_0000h	42.3.2/1155
F800_0048	Port Clear Output Register (FGPIOB_PCOR)	32	W (always reads 0)	0000_0000h	42.3.3/1156
F800_004C	Port Toggle Output Register (FGPIOB_PTOR)	32	W (always reads 0)	0000_0000h	42.3.4/1156
F800_0050	Port Data Input Register (FGPIOB_PDIR)	32	R	0000_0000h	42.3.5/1157
F800_0054	Port Data Direction Register (FGPIOB_PDDR)	32	R/W	0000_0000h	42.3.6/1157
F800_0080	Port Data Output Register (FGPIOC_PDOR)	32	R/W	0000_0000h	42.3.1/1155
F800_0084	Port Set Output Register (FGPIOC_PSOR)	32	W (always reads 0)	0000_0000h	42.3.2/1155
F800_0088	Port Clear Output Register (FGPIOC_PCOR)	32	W (always reads 0)	0000_0000h	42.3.3/1156
F800_008C	Port Toggle Output Register (FGPIOC_PTOR)	32	W (always reads 0)	0000_0000h	42.3.4/1156
F800_0090	Port Data Input Register (FGPIOC_PDIR)	32	R	0000_0000h	42.3.5/1157
F800_0094	Port Data Direction Register (FGPIOC_PDDR)	32	R/W	0000_0000h	42.3.6/1157

42.3.1 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.

42.3.2 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.

42.3.3 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.

42.3.4 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.

42.3.5 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.

42.3.6 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.

42.4 Functional description

42.4.1 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.

42.4.2 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.

42.4.3 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 43
LP Trusted Cryptography (LTC)

43.1 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.

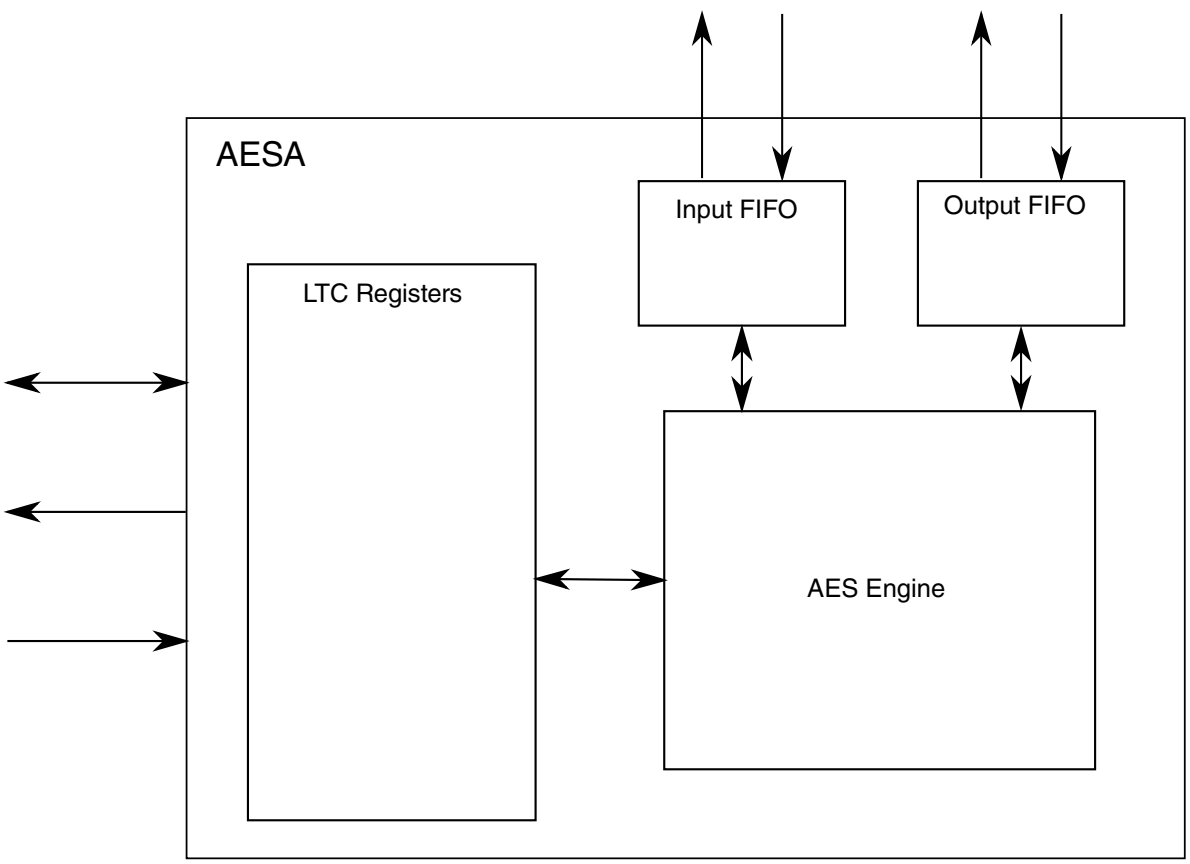


Figure 43-1. LTC Block Diagram

43.2 Feature summary

LTC includes the following actives:

- Cryptographic authentication
  - Message authentication codes (MAC)
    - AES-CMAC
    - AES-XCBC-MAC
  - Auto padding
  - ICV checking
- Authenticated encryption algorithms
  - AES-CCM (counter with CBC-MAC)
- Symmetric key block ciphers
  - AES (128-bit keys)
  - Cipher modes
    - ECB, CBC, CTR for AES
- Secure Scan

43.3 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.

43.3.1 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.

43.3.2 AESA modes of operation

The following modes are supported by AESA:

- Electronic codebook (ECB)
- Cipher block chaining (CBC)
- Counter (CTR)
- Extended cipher block chaining message authentication code (XCBC-MAC)
- Cipher-based MAC (CMAC)
- CTR and CBC-MAC (CCM)

AES modes can be classified into these categories:

- Confidentiality (ECB, CBC, CTR)
- Authenticated Confidentiality (CCM, CCM*)
- 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.

43.3.3 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.

43.3.4 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.

43.3.4.1 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.

43.3.4.2 AES ECB mode use of the Context Register

ECB does not use Context Registers.

43.3.4.3 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.

43.3.4.4 AES ECB Mode use of the Key Register

ECB keys must be written to the Key Register and can have only 16 bytes.

43.3.4.5 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. Any value other than 16 causes the key-size error to be generated.

43.3.5 AES CBC mode

The CBC mode is described in this table.

Table 43-1. 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.

43.3.5.1 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.

43.3.5.2 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 43-2. Context usage in CBC mode

Context Word	Definition
0	IV [127:96]
1	IV [95:64]
2	IV [63:32]
3	IV [31:0]

43.3.5.3 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.

43.3.5.4 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.

43.3.5.5 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.
- Any value other than 16 causes a key-size error to be generated.

43.3.6 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. CAAM 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.

43.3.6.1 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.

43.3.6.2 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 43-3. 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]

43.3.6.3 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.

43.3.6.4 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.

43.3.6.5 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. Any value other than 16 will cause Key Size error to be generated.

43.3.7 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.

43.3.7.1 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 43-4. 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 43-5. 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.

43.3.7.2 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 43-6. 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]
	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]

Table continues on the next page...

Table 43-6. Context usage in XCBC-MAC and CMAC modes (continued)

Mode	Context word	Context-switching definition	Final-result definition
	4	L[127:96]	-
	5	L[95:64]	-
	6	L[63:32]	-
	7	L[31:0]	-

43.3.7.3 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.

43.3.7.4 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.

43.3.7.5 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.

43.3.7.6 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.
- For XCBC-MAC, any value other than 16 causes a key-size error to be generated. For CMAC, this error is generated only if any value other than 16 is written.

43.3.7.7 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.

43.3.8 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.

43.3.8.1 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.

43.3.8.2 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.

43.3.8.3 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 43-7. 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.

43.3.8.4 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 43-8. 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 43-10	-
13	-	MAC size; see Table 43-11	-

1. E(x) means encrypted x

Table 43-9. 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]	-
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 43-10	-
13	-	MAC size; see Table 43-11	-

Table 43-10. 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 43-11. Format of Context Word 13 for AES-CCM and AES-CCM* mode

Bits 31-3	Bits 2-0
0	Encoded MAC Size

43.3.8.5 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.

43.3.8.6 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.

43.3.8.7 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.
- Any value other than 16 causes a key-size error to be generated.

43.3.8.8 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.

43.4 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.

43.5 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.

43.6 LTC register descriptions

All reads of write-only addresses always return zero. Writes to read-only addresses are ignored. LTC will generate a transfer error whenever an undefined address is read or written to 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.

43.6.1 LTC Memory map

LTC0 base address: 4005_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Mode Register (MD)	32	RW	0000_0000h
8h	Key Size Register (KS)	32	WO	0000_0010h
10h	Data Size Register (DS)	32	RW	0000_0000h
18h	ICV Size Register (ICVS)	32	RW	0000_0000h
30h	Command Register (COM)	32	WO	0000_0000h
34h	Control Register (CTL)	32	RW	0000_0000h
40h	Clear Written Register (CW)	32	WO	0000_0000h
48h	Status Register (STA)	32	W1C	0000_0000h
4Ch	Error Status Register (ESTA)	32	RO	0000_0000h
58h	AAD Size Register (AADSZ)	32	RW	0000_0000h
100h - 134h	Context Register (CTX_0 - CTX_13)	32	RW	0000_0000h
200h - 20Ch	Key Registers (KEY_0 - KEY_3)	32	RW	0000_0000h
4F0h	Version ID Register (VID1)	32	RO	0034_0100h
4F4h	Version ID 2 Register (VID2)	32	RO	0000_0101h
4F8h	CHA Version ID Register (CHAVID)	32	RO	0000_0050h
7C0h	FIFO Status Register (FIFOSTA)	32	RO	0000_0000h
7E0h	Input Data FIFO (IFIFO)	32	WO	0000_0000h
7F0h	Output Data FIFO (OFIFO)	32	RO	0000_0000h

43.6.2 Mode Register (MD)

43.6.2.1 Offset

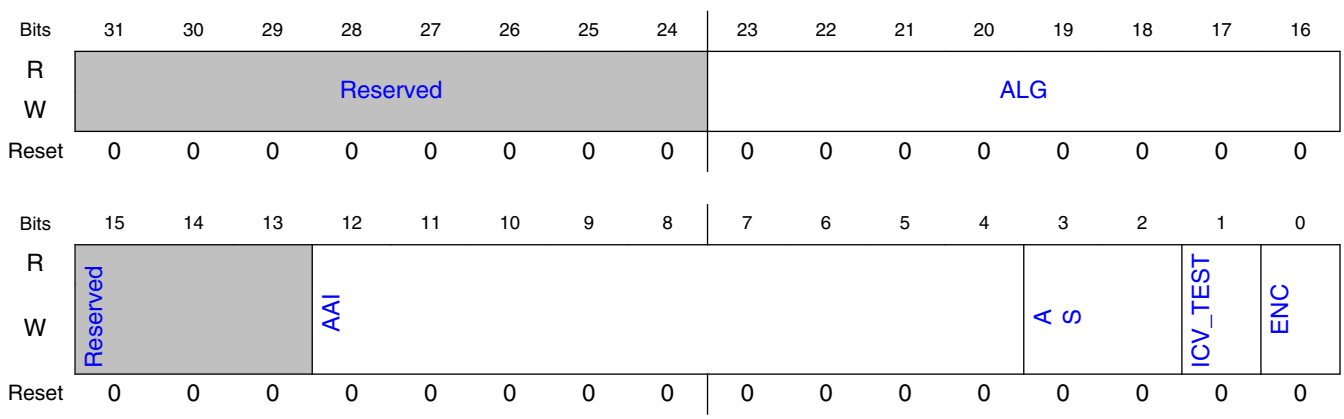
Register	Offset
MD	0h

43.6.2.2 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.

43.6.2.3 Diagram



43.6.2.4 Fields

Field	Function
31-24 —	Reserved. Must be 0.
23-16 ALG	Algorithm This field specifies which algorithm is being selected. 00010000b - AES
15-13 —	Reserved. Must be 0.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></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.
NOTE: Some algorithms do not require additional algorithm information and in those cases this field should be all 0s.
Table 43-12. AAI Interpretation for AES Modes
<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>CTR_XCBC_MAC</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 colspan="3"></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.
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</td></tr>
<tr><td>0</td><td>Encrypt/Decrypt.</td></tr>
</table>

Field	Function
ENC	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).

43.6.3 Key Size Register (KS)

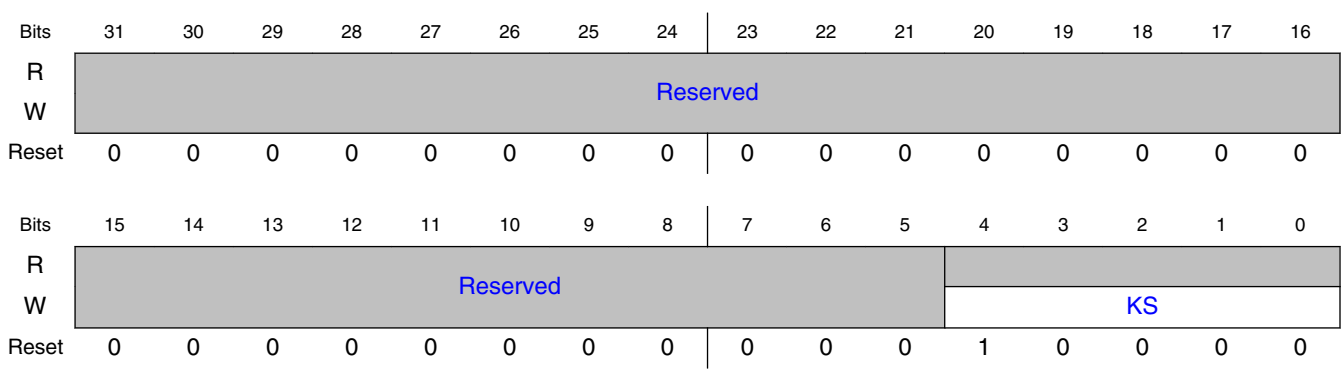
43.6.3.1 Offset

Register	Offset
KS	8h

43.6.3.2 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.

43.6.3.3 Diagram



43.6.3.4 Fields

Field	Function
31-5 —	Reserved.
4-0 KS	Key Size This is the size of a Key measured in bytes

43.6.4 Data Size Register (DS)

43.6.4.1 Offset

Register	Offset
DS	10h

43.6.4.2 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 AAD Size Register (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.

43.6.4.3 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

43.6.4.4 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.

43.6.5 ICV Size Register (ICVS)

43.6.5.1 Offset

Register	Offset
ICVS	18h

43.6.5.2 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.

43.6.5.3 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

43.6.5.4 Fields

Field	Function
31-5 —	Reserved.
4-0 ICVS	ICV Size, in Bytes

43.6.6 Command Register (COM)

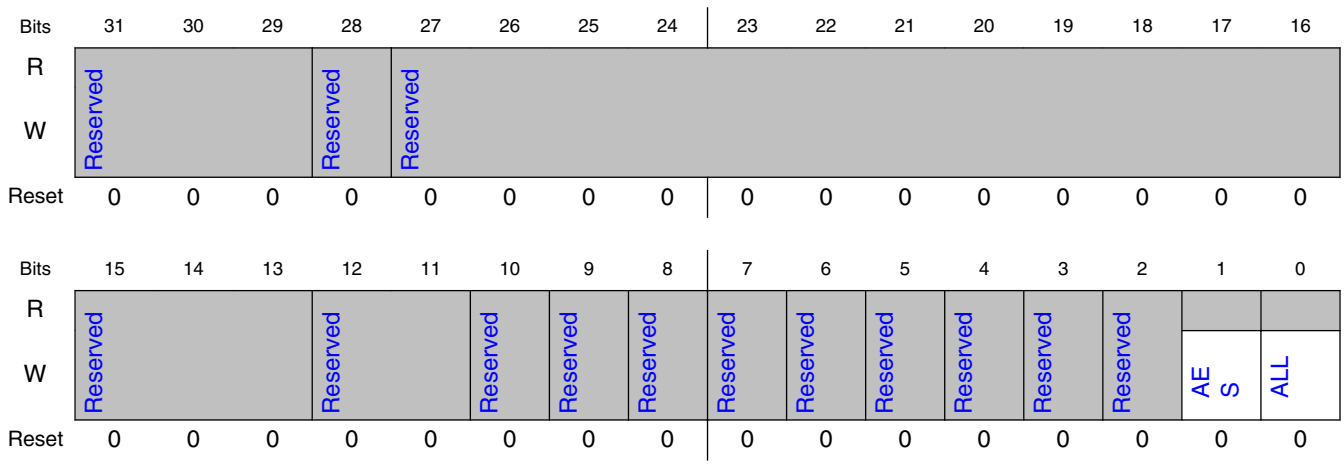
43.6.6.1 Offset

Register	Offset
COM	30h

43.6.6.2 Function

The Command Register is used to send control signals to the Crypto Engines.

43.6.6.3 Diagram



43.6.6.4 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.

Table continues on the next page...

Field	Function
6 —	Reserved.
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.

43.6.7 Control Register (CTL)

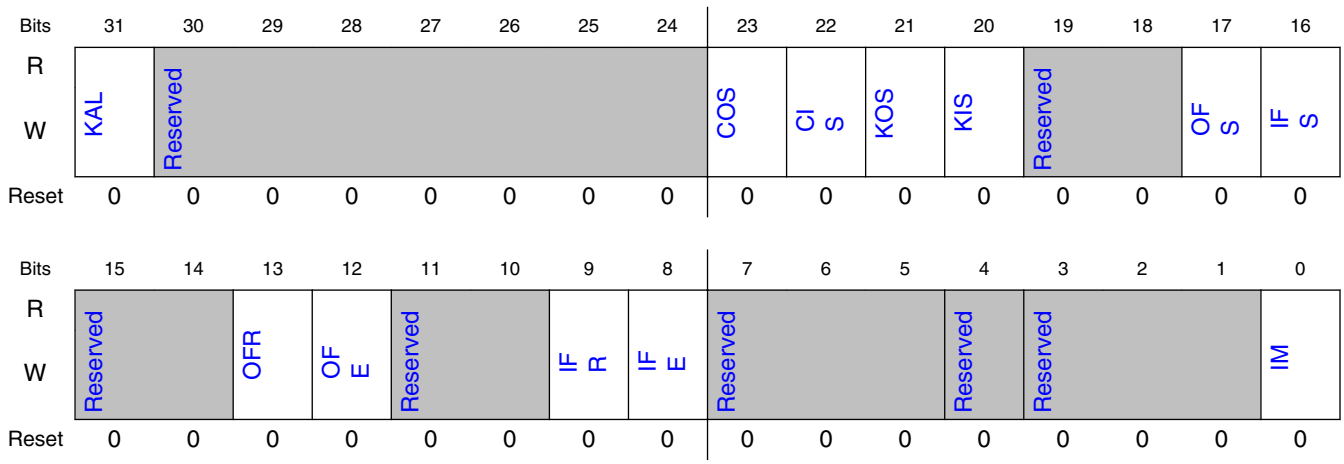
43.6.7.1 Offset

Register	Offset
CTL	34h

43.6.7.2 Function

This register is used for some of the internal controls of the LTC block.

43.6.7.3 Diagram



43.6.7.4 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.

Table continues on the next page...

Field	Function
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.
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

43.6.8 Clear Written Register (CW)

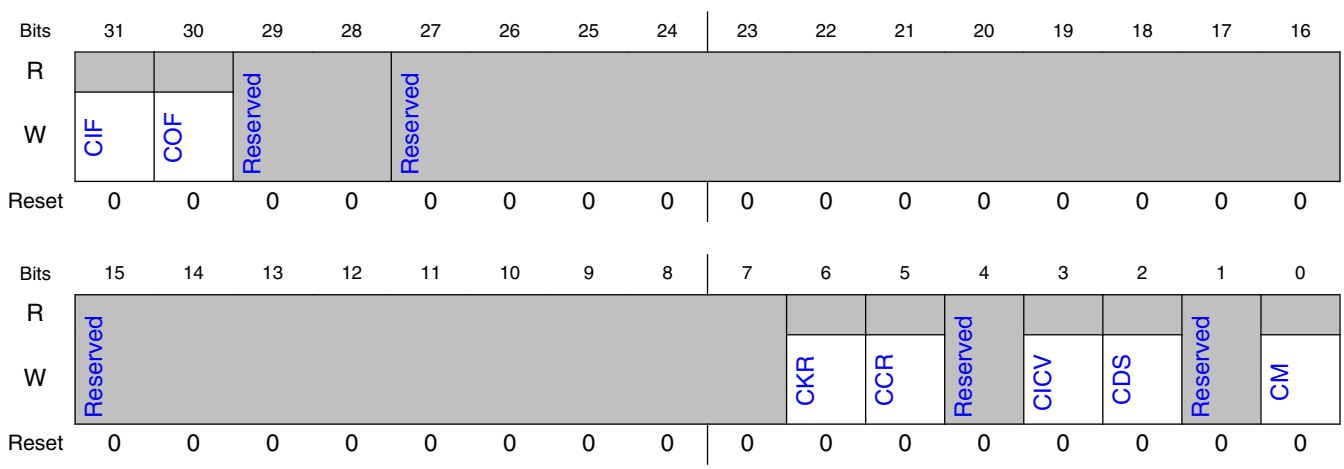
43.6.8.1 Offset

Register	Offset
CW	40h

43.6.8.2 Function

The Clear Written Register is used to clear many of the internal registers. All fields of this register are self-clearing.

43.6.8.3 Diagram



43.6.8.4 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.

Table continues on the next page...

Field	Function
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.
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.

43.6.9 Status Register (STA)

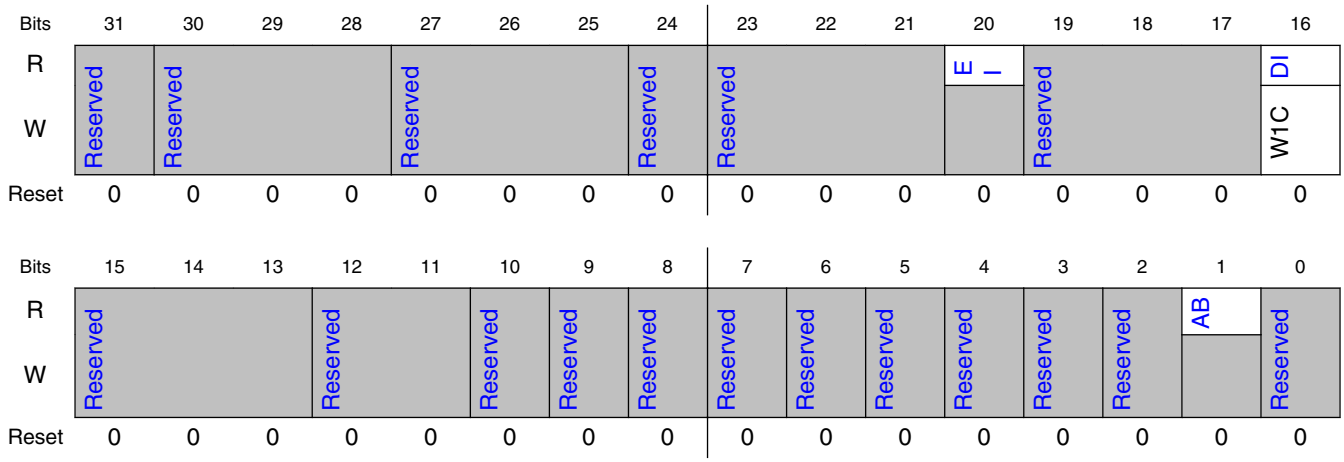
43.6.9.1 Offset

Register	Offset
STA	48h

43.6.9.2 Function

The Status Register shows the status of the internal Crypto engine and its internal registers.

43.6.9.3 Diagram



43.6.9.4 Fields

Field	Function
31 —	Reserved.
30-28 —	Reserved.
27-25 —	Reserved.
24 —	Reserved.
23-21 —	Reserved.
20 EI	Error Interrupt The Error Interrupt has been asserted. This error can only be cleared by resetting LTC. 0b - Not Error. 1b - Error Interrupt.
19-17 —	Reserved.
16 DI	Done Interrupt The Done Interrupt has been asserted.

Value	Read	Write
0	No Done Interrupt	No change

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td><table>
<tr><th>Value</th><th>Read</th><th>Write</th></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>
<tr><td>4
—</td><td>Reserved.</td></tr>
<tr><td>3
—</td><td>Reserved.</td></tr>
<tr><td>2
—</td><td>Reserved.</td></tr>
<tr><td>1
AB</td><td>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.</td></tr>
<tr><td>0
—</td><td>Reserved.</td></tr>
</table>

43.6.10 Error Status Register (ESTA)

43.6.10.1 Offset

Register	Offset
ESTA	4Ch

43.6.10.2 Function

The Error Register shows the status of the internal Crypto Engine and its internal registers.

43.6.10.3 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

43.6.10.4 Fields

Field	Function
31-12 —	Reserved.
11-8 CL1	algorithms The algorithms field indicates which algorithm is asserting an error. Others reserved 0000b - General Error 0001b - AES
7-4 —	Reserved.
3-0 ERRID1	Error ID 1

Field	Function
	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

43.6.11 AAD Size Register (AADSZ)

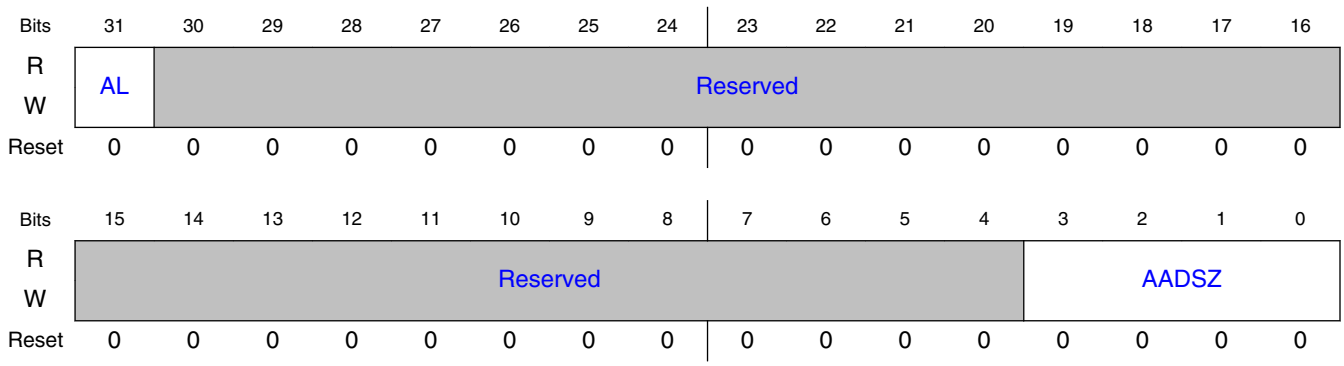
43.6.11.1 Offset

Register	Offset
AADSZ	58h

43.6.11.2 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 AAD Size Register (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.

43.6.11.3 Diagram



43.6.11.4 Fields

Field	Function
31 AL	AAD Last Only AAD data will be written into the Input FIFO.
30-4 —	Reserved.
3-0 AADSZ	AAD size in Bytes, mod 16

43.6.12 Context Register (CTX_0 - CTX_13)

43.6.12.1 Offset

For a = 0 to 13:

Register	Offset
CTX_a	100h + (a × 4h)

43.6.12.2 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 and CCM* modes use of the Context Register

43.6.12.3 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

43.6.12.4 Fields

Field	Function
31-0 CTX	CTX

43.6.13 Key Registers (KEY_0 - KEY_3)

43.6.13.1 Offset

For a = 0 to 3:

Register	Offset
KEY_a	200h + (a × 4h)

43.6.13.2 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.

43.6.13.3 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

43.6.13.4 Fields

Field	Function
31-0 KEY	KEY

43.6.14 Version ID Register (VID1)

43.6.14.1 Offset

Register	Offset
VID1	4F0h

43.6.14.2 Function

This register contains the ID for LTC and major and minor revision numbers.

43.6.14.3 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

43.6.14.4 Fields

Field	Function
31-16 IP_ID	ID(0x0034).
15-8 MAJ_REV	Major revision number.
7-0 MIN_REV	Minor revision number.

43.6.15 Version ID 2 Register (VID2)

43.6.15.1 Offset

Register	Offset
VID2	4F4h

43.6.15.2 Function

This register contains the architectural era and eco revision numbers.

43.6.15.3 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

43.6.15.4 Fields

Field	Function
31-16 —	Reserved.
15-8 ARCH_ERA	Architectural ERA.
7-0 ECO_REV	ECO revision number.

43.6.16 CHA Version ID Register (CHAVID)

43.6.16.1 Offset

Register	Offset
CHAVID	4F8h

43.6.16.2 Function

This register contains the Version ID and Revision Number for the CHAs contained within LTC.

43.6.16.3 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

43.6.16.4 Fields

Field	Function
31-24 —	Reserved.
23-16 —	Reserved.
15-8 —	Reserved.

Table continues on the next page...

Field	Function
7-4 AESVID	AES Version ID
3-0 AESREV	AES Revision Number

43.6.17 FIFO Status Register (FIFOSTA)

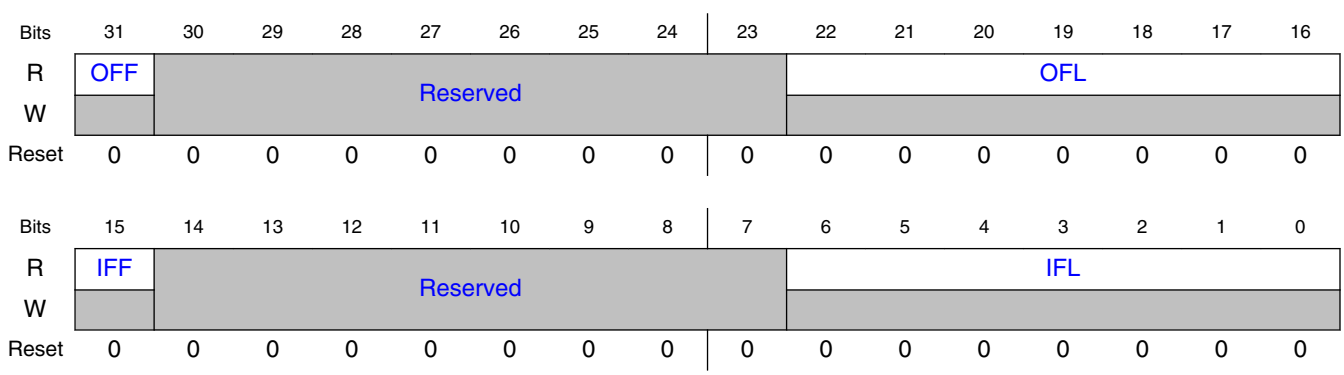
43.6.17.1 Offset

Register	Offset
FIFOSTA	7C0h

43.6.17.2 Function

The FIFO Status shows the current levels of the Input and Output FIFO.

43.6.17.3 Diagram



43.6.17.4 Fields

Field	Function
31	Output FIFO Full

Table continues on the next page...

Field	Function
OFF	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.

43.6.18 Input Data FIFO (IFIFO)

43.6.18.1 Offset

Register	Offset
IFIFO	7E0h

43.6.18.2 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.

43.6.18.3 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

43.6.18.4 Fields

Field	Function
31-0 IFIFO	IFIFO

43.6.19 Output Data FIFO (OFIFO)

43.6.19.1 Offset

Register	Offset
OFIFO	7F0h

43.6.19.2 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.

43.6.19.3 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

43.6.19.4 Fields

Field	Function
31-0 OFIFO	Output FIFO

Chapter 44
True Random Number Generator (TRNG)

44.1 Standalone True Random Number Generator (SA-TRNG).

The Standalone True Random Number Generator (SA-TRNG) is a 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) that 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 a 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).

44.1.1 Standalone True Random Number Generator Block Diagram

The following figure is a top-level diagram of the True Random Number Generator.

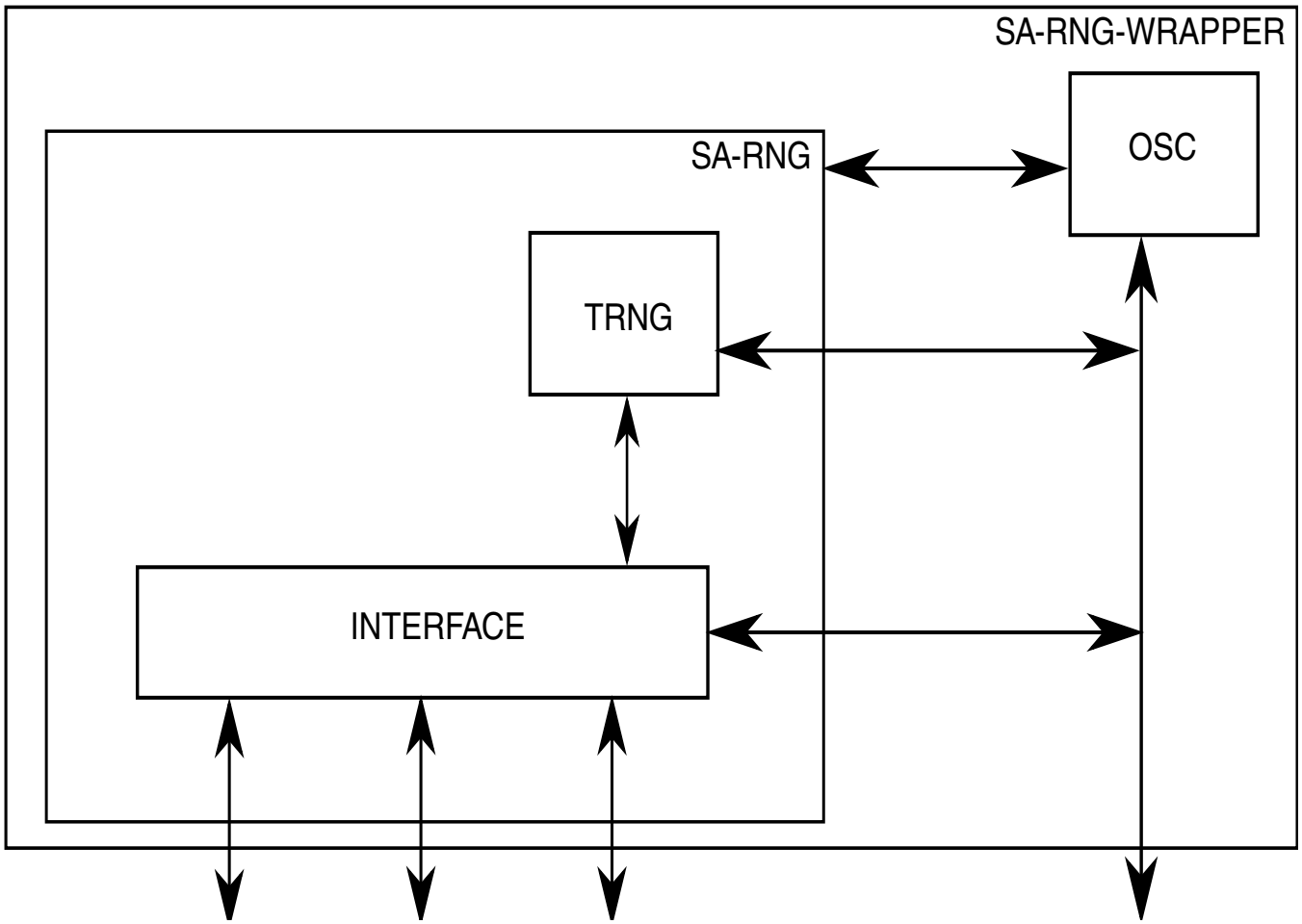


Figure 44-1. SA TRNG Block Diagram

44.1.2 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.

44.1.3 SA-TRNG hardware functional description.

SA-TRNG functionality consists of several major subcomponents. This table describes these subcomponents.

Table 44-1. 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).

44.1.3.1 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.

44.1.3.2 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 44-2. 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 •

NOTE

Accesses to registers must use full-word (32-bit) reads or writes.

44.1.4 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.

44.1.5 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).

44.1.5.1 TRNG0 Memory map

TRNG0 base address: 4002_9000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Miscellaneous Control Register (MCTL)	32	RW	0001_2001h
4h	Statistical Check Miscellaneous Register (SCMISC)	32	RW	0001_001Fh
8h	Poker Range Register (PKRRNG)	32	RW	0000_09A3h
Ch	Poker Maximum Limit Register (PKRMAX)	32	RW	0000_6920h
Ch	Poker Square Calculation Result Register (PKRSQ)	32	RO	0000_0000h
10h	Seed Control Register (SDCTL)	32	RW	0C80_09C4h
14h	Sparse Bit Limit Register (SBLIM)	32	RW	0000_003Fh
14h	Total Samples Register (TOTSAM)	32	RO	0000_0000h
18h	Frequency Count Minimum Limit Register (FRQMIN)	32	RW	0000_0640h
1Ch	Frequency Count Register (FRQCNT)	32	RO	0000_0000h
1Ch	Frequency Count Maximum Limit Register (FRQMAX)	32	RW	0000_6400h
20h	Statistical Check Monobit Count Register (SCMC)	32	RO	0000_0000h
20h	Statistical Check Monobit Limit Register (SCML)	32	RW	010C_0568h
24h	Statistical Check Run Length 1 Count Register (SCR1C)	32	RO	0000_0000h
24h	Statistical Check Run Length 1 Limit Register (SCR1L)	32	RW	00B2_0195h
28h	Statistical Check Run Length 2 Count Register (SCR2C)	32	RO	0000_0000h
28h	Statistical Check Run Length 2 Limit Register (SCR2L)	32	RW	007A_00DCh
2Ch	Statistical Check Run Length 3 Count Register (SCR3C)	32	RO	0000_0000h
2Ch	Statistical Check Run Length 3 Limit Register (SCR3L)	32	RW	0058_007Dh
30h	Statistical Check Run Length 4 Count Register (SCR4C)	32	RO	0000_0000h
30h	Statistical Check Run Length 4 Limit Register (SCR4L)	32	RW	0040_004Bh
34h	Statistical Check Run Length 5 Count Register (SCR5C)	32	RO	0000_0000h
34h	Statistical Check Run Length 5 Limit Register (SCR5L)	32	RW	002E_002Fh
38h	Statistical Check Run Length 6+ Count Register (SCR6PC)	32	RO	0000_0000h
38h	Statistical Check Run Length 6+ Limit Register (SCR6PL)	32	RW	002E_002Fh
3Ch	Status Register (STATUS)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
40h - 7Ch	Entropy Read Register (ENT0 - ENT15)	32	RO	0000_0000h
80h	Statistical Check Poker Count 1 and 0 Register (PKRCNT10)	32	RO	0000_0000h
84h	Statistical Check Poker Count 3 and 2 Register (PKRCNT32)	32	RO	0000_0000h
88h	Statistical Check Poker Count 5 and 4 Register (PKRCNT54)	32	RO	0000_0000h
8Ch	Statistical Check Poker Count 7 and 6 Register (PKRCNT76)	32	RO	0000_0000h
90h	Statistical Check Poker Count 9 and 8 Register (PKRCNT98)	32	RO	0000_0000h
94h	Statistical Check Poker Count B and A Register (PKRCNTBA)	32	RO	0000_0000h
98h	Statistical Check Poker Count D and C Register (PKRCNTDC)	32	RO	0000_0000h
9Ch	Statistical Check Poker Count F and E Register (PKRCNTFE)	32	RO	0000_0000h
B0h	Security Configuration Register (SEC_CFG)	32	RW	0000_0000h
B4h	Interrupt Control Register (INT_CTRL)	32	RW	FFFF_FFFFh
B8h	Mask Register (INT_MASK)	32	RW	0000_0000h
BCh	Interrupt Status Register (INT_STATUS)	32	RO	0000_0000h
F0h	Version ID Register (MS) (VID1)	32	RO	0030_0100h
F4h	Version ID Register (LS) (VID2)	32	RO	0000_0000h

44.1.5.2 Miscellaneous Control Register (MCTL)

44.1.5.2.1 Offset

Register	Offset
MCTL	0h

44.1.5.2.2 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 bit below.

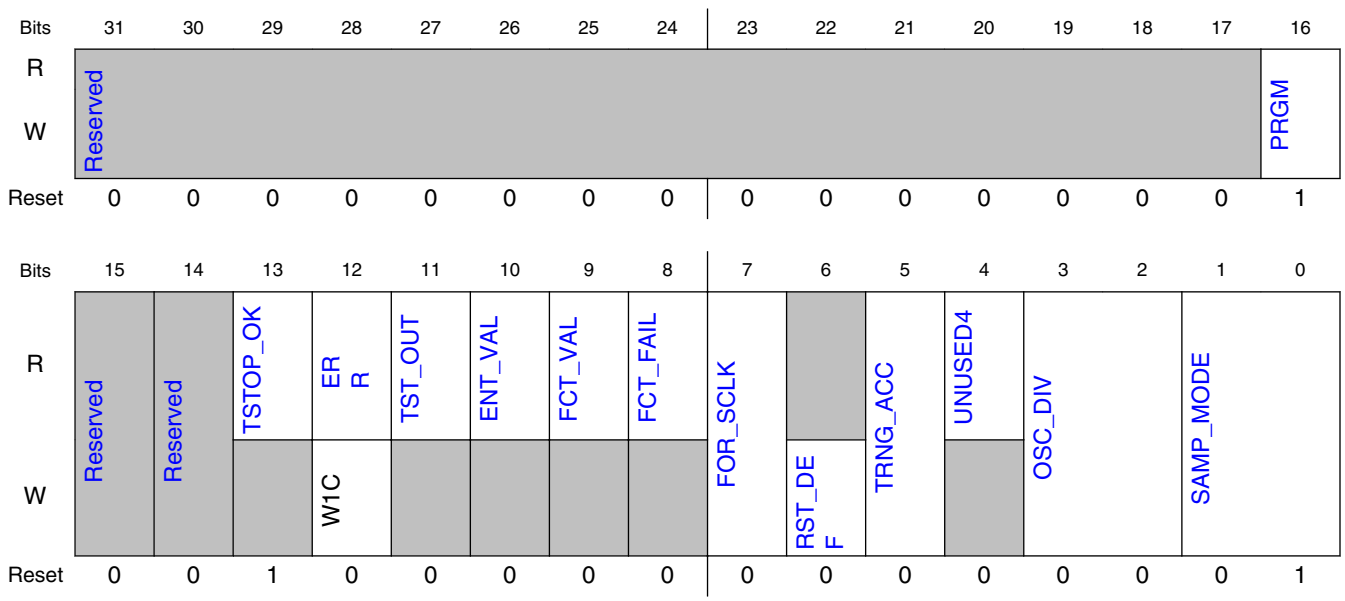
The Miscellaneous Control Register is a read/write register used to control the RNG's True Random Number Generator (TRNG) access, operation and test.

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 MCTL register.

44.1.5.2.3 Diagram



44.1.5.2.4 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 —	Reserved.
14 —	Reserved.
13 TSTOP_OK	TRNG_OK_TO_STOP. Software can 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. To make sure the ring oscillator is stopped, assert ipg_stop input and verify ipg_rng_stop_ack is asserted after at at least two (2) clock cycles. TSTOP_OK is asserted when the TRNG ring oscillator is not running. This helps for cases where you want to stop the TRNG0 TRNG clock without having access to ipg_stop nor ipg_rng_stop_ack.
12	Read: Error status. 1 = error detected. 0 = no error.

Table continues on the next page...

Field	Function
ERR	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 ENT15 register. (ENT0 through ENT14 should be read before reading ENT15).
9 FCT_VAL	Read only: Frequency Count Valid. Indicates that a valid frequency count may be read from 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 FRQMAX and/or FRQMIN registers, or a hardware failure in the ring oscillator. This error may be cleared by writing a 1 to the ERR bit.
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 ENT0-ENT15 registers. The Entropy value may be read once the ENT VAL bit is asserted. Also see ENTa register descriptions (For a = 0 to 15).
4 UNUSED4	This bit is unused. Always reads 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. 00b - use ring oscillator with no divide 01b - use ring oscillator divided-by-2 10b - use ring oscillator divided-by-4 11b - 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. 00b - use Von Neumann data into both Entropy shifter and Statistical Checker 01b - use raw data into both Entropy shifter and Statistical Checker 10b - use Von Neumann data into Entropy shifter. Use raw data into Statistical Checker 11b - undefined/reserved.

44.1.5.3 Statistical Check Miscellaneous Register (SCMISC)

44.1.5.3.1 Offset

Register	Offset
SCMISC	4h

44.1.5.3.2 Function

The Statistical Check Miscellaneous Register contains the Long Run Maximum Limit value and the Retry Count value. This register is accessible only when the MCTL[PRGM] bit is 1, otherwise this register will read zeroes, and cannot be written.

44.1.5.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												RTY_CT			
Reset [1]	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 W	Reserved								LRUN_MAX							
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

1. Reset occurs at POR, and when MCTL[RST_DEF] is written to 1.

44.1.5.3.4 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 MCTL[PRGM] bit is 1. This field will read zeroes if MCTL[PRGM] = 0. This field is cleared to the default POR value by writing the 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 MCTL[PRGM] bit is 1. This field will read zeroes if MCTL[PRGM] = 0. This field is cleared to the POR reset value by writing the MCTL[RST_DEF] bit to 1.

44.1.5.4 Poker Range Register (PKRRNG)

44.1.5.4.1 Offset

Register	Offset
PKRRNG	8h

44.1.5.4.2 Function

The 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.

44.1.5.4.3 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

44.1.5.4.4 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 MCTL[PRGM] bit is 1. This field will read zeroes if MCTL[PRGM] = 0. This field is cleared to the default POR value by writing the MCTL[RST_DEF] bit to 1. Note that the minimum allowable Poker result is PKR_MAX - PKR_RNG + 1.

44.1.5.5 Poker Square Calculation Result Register (PKRSQ)

44.1.5.5.1 Offset

Register	Offset	Description
PKRSQ	Ch	Accessible at this address when MCTL[PRGM] = 0]

44.1.5.5.2 Function

The 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 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 PKRRNG value. Note that this offset (0x0C) is used as PKRMAX if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as PKRSQ readback register, as described here.

44.1.5.5.3 Diagram

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

44.1.5.5.4 Fields

Field	Function
31-24 —	Reserved.
23-0 PKR_SQ	Poker Square Calculation Result. During the TRNG Statistical Checks, a "Poker Test" is run which starts with the value PKRMAX[PKR_MAX]. This value decreases according to a "sum of squares" algorithm, and must remain greater than zero, but less than the PKRRNG[PKR_RNG] limit. The resulting value may be read through this register, if MCTL[PRGM] bit is 0. Note that if MCTL[PRGM] bit is 1, this register address is used to access the Poker Test Maximum Limit in register PKRMAX, as defined in the previous section.

44.1.5.6 Poker Maximum Limit Register (PKRMAX)

44.1.5.6.1 Offset

Register	Offset	Description
PKRMAX	Ch	Accessible at this address when MCTL[PRGM] = 1]

44.1.5.6.2 Function

The Poker Maximum Limit Register defines Maximum Limit allowable during the TRNG Statistical Check Poker Test. Note that this offset (0x0C) is used as PKRMAX only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as the PKRSQ readback register.

44.1.5.6.3 Diagram

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

44.1.5.6.4 Fields

Field	Function
31-24 —	Reserved.
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 PKRMAX[PKR_MAX] register. This field is writable only if MCTL[PRGM] bit is 1. This register is cleared to the default POR value by writing the MCTL[RST_DEF] bit to 1. Note that the PKRMAX and PKRRNG registers combined are used to define the minimum allowable Poker result, which is PKR_MAX - PKR_RNG + 1. Note that if MCTL[PRGM] bit is 0, this register address is used to read the Poker Test Square Calculation result in register PKRSQ, as defined in the following section.

44.1.5.7 Seed Control Register (SDCTL)

44.1.5.7.1 Offset

Register	Offset
SDCTL	10h

44.1.5.7.2 Function

The 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).

44.1.5.7.3 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

44.1.5.7.4 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 MCTL[PRGM] bit is 1. This field will read zeroes if 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 MCTL[PRGM] bit is 1. This field will read zeroes if MCTL[PRGM] = 0. This field is cleared to the default POR value by writing the MCTL[RST_DEF] bit to 1.

44.1.5.8 Sparse Bit Limit Register (SBLIM)

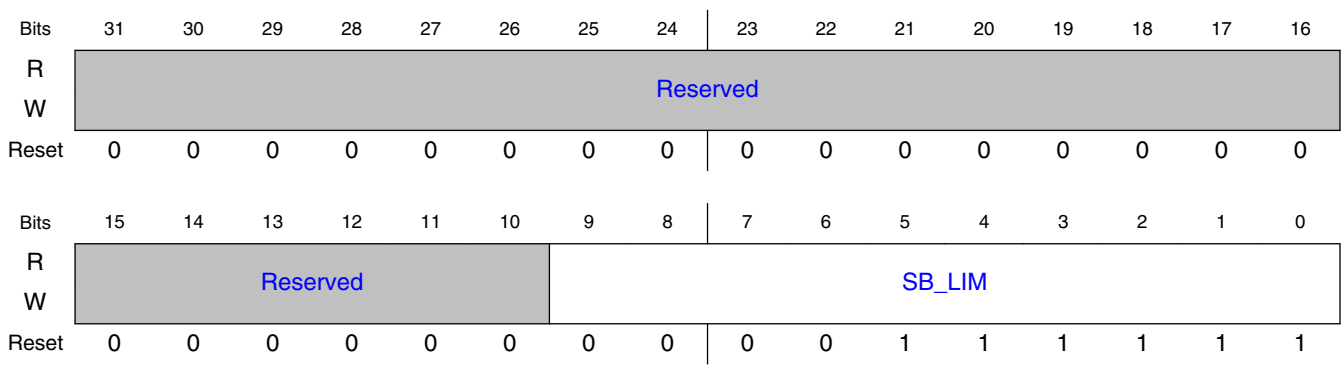
44.1.5.8.1 Offset

Register	Offset	Description
SBLIM	14h	Accessible at this address when MCTL[PRGM] = 1]

44.1.5.8.2 Function

The 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 SBLIM only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as TOTSAM readback register.

44.1.5.8.3 Diagram



44.1.5.8.4 Fields

Field	Function
31-10 —	Reserved. Always 0.
9-0 SB_LIM	Sparse Bit Limit. During Von Neumann sampling (if enabled by 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 MCTL[PRGM] bit is 1. This register is cleared to the default POR value by writing the MCTL[RST_DEF] bit to 1. Note that if MCTL[PRGM] bit is 0, this register address is used to read the Total Samples count in register TOTSAM, as defined in the following section.

44.1.5.9 Total Samples Register (TOTSAM)

44.1.5.9.1 Offset

Register	Offset	Description
TOTSAM	14h	Accessible at this address when MCTL[PRGM] = 0]

44.1.5.9.2 Function

The 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 SBLIM if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as TOTSAM readback register, as described here.

44.1.5.9.3 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

44.1.5.9.4 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 MCTL[PRGM] bit is 0. Note that if MCTL[PRGM] bit is 1, this register address is used to access the Sparse Bit Limit in register SBLIM, as defined in the previous section.

44.1.5.10 Frequency Count Minimum Limit Register (FRQMIN)

44.1.5.10.1 Offset

Register	Offset
FRQMIN	18h

44.1.5.10.2 Function

The 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 MCTL[FCT_FAIL] and an error is generated.

44.1.5.10.3 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

44.1.5.10.4 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 MCTL[PRGM] bit is 1. This field will read zeroes if MCTL[PRGM] = 0. This field is cleared to its reset value at POR.

44.1.5.11 Frequency Count Maximum Limit Register (FRQMAX)

44.1.5.11.1 Offset

Register	Offset	Description
FRQMAX	1Ch	Accessible at this address when MCTL[PRGM] = 1]

44.1.5.11.2 Function

The 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 MCTL[FCT_FAIL] and an error is generated. Note that this address (001C) is used as FRQMAX only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as FRQCNT readback register.

44.1.5.11.3 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

44.1.5.11.4 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 MCTL[PRGM] bit is 1. This field is cleared to its reset value at POR. Note that if MCTL[PRGM] bit is 0, this register address is used to read the Frequency Count result in register FRQCNT, as defined in the FRQ_CT field.

44.1.5.12 Frequency Count Register (FRQCNT)

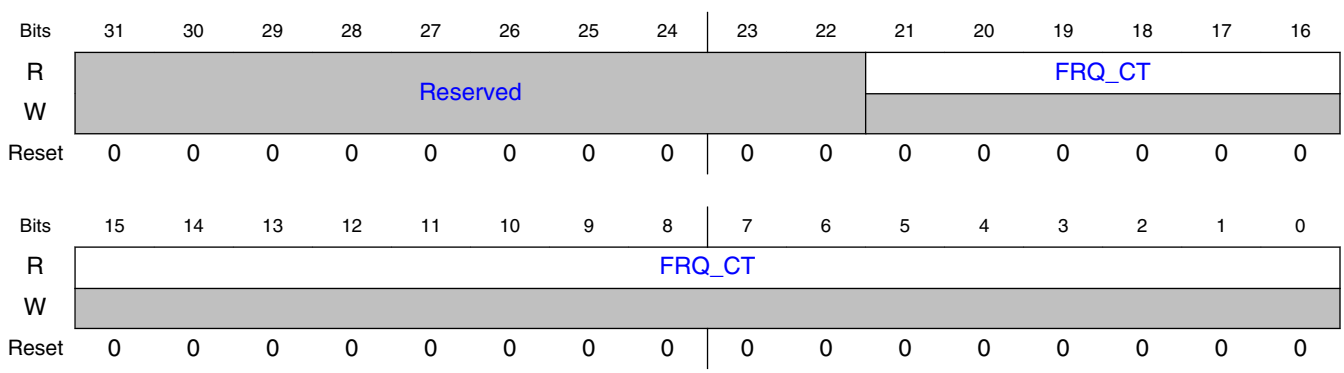
44.1.5.12.1 Offset

Register	Offset	Description
FRQCNT	1Ch	Accessible at this address when MCTL[PRGM] = 0]

44.1.5.12.2 Function

The 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 PRGM[MCTL] = 0 and MCTL[TRNG_ACC] = 1. Note that this offset (0x1C) is used as FRQMAX if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as live FRQCNT readback register, as described here. This register must be read once, as a full 32-bit. A subsequent read is only valid when MCTL[FCT_VAL] bit is HIGH. A read clears this register. The next read when MCTL[FCT_VAL] bit is HIGH must be treated as a different value from the previous one, even though the counts might be the same.

44.1.5.12.3 Diagram



44.1.5.12.4 Fields

Field	Function
31-22 —	Reserved. Always 0.
21-0 FRQ_CT	Frequency Count. If MCTL[TRNG_ACC] = 1, Reads a sample frequency count taken during entropy generation. Requires MCTL[PRGM] = 0. Note that if MCTL[PRGM] bit is 1, this register address is used to access the Frequency Maximum Limit in register FRQMAX, as defined in the FRQ_MAX field.

44.1.5.13 Statistical Check Monobit Count Register (SCMC)

44.1.5.13.1 Offset

Register	Offset	Description
SCMC	20h	Accessible at this address when MCTL[PRGM] = 0]

44.1.5.13.2 Function

The 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 SCML[MONO_MAX], and is decremented each time a one is sampled. Note that this offset (0x20) is used as SCML if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCMC readback register, as described here.

44.1.5.13.3 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

44.1.5.13.4 Fields

Field	Function
31-16 —	Reserved. Always 0.
15-0 MONO_CT	Monobit Count. Reads the final Monobit count after entropy generation. Requires MCTL[PRGM] = 0. Note that if MCTL[PRGM] bit is 1, this register address is used to access the Statistical Check Monobit Limit in register SCML, as defined in the previous section.

44.1.5.14 Statistical Check Monobit Limit Register (SCML)

44.1.5.14.1 Offset

Register	Offset	Description
SCML	20h	Accessible at this address when MCTL[PRGM] = 1]

44.1.5.14.2 Function

The 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 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 SCML only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCMC readback register.

44.1.5.14.3 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

44.1.5.14.4 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 MCTL[RST_DEF] bit to 1.
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 MCTL[RST_DEF] bit to 1.

44.1.5.15 Statistical Check Run Length 1 Limit Register (SCR1L)

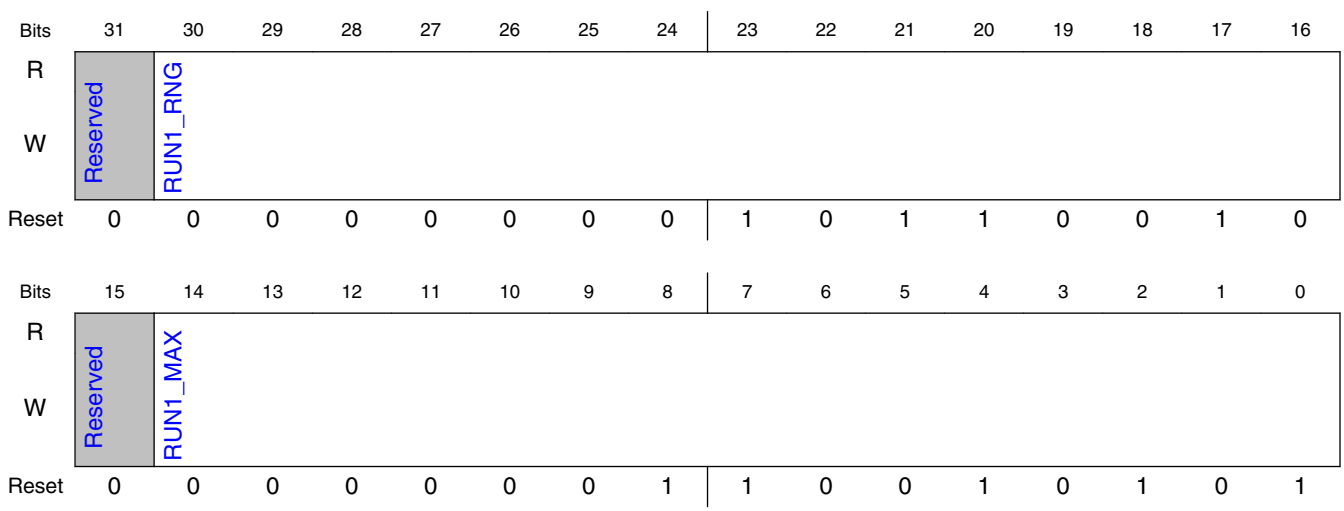
44.1.5.15.1 Offset

Register	Offset	Description
SCR1L	24h	Accessible at this address when MCTL[PRGM] = 1]

44.1.5.15.2 Function

The 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 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 SCR1L only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as SCR1C readback register.

44.1.5.15.3 Diagram



44.1.5.15.4 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 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 MCTL[RST_DEF] bit to 1.

44.1.5.16 Statistical Check Run Length 1 Count Register (SCR1C)

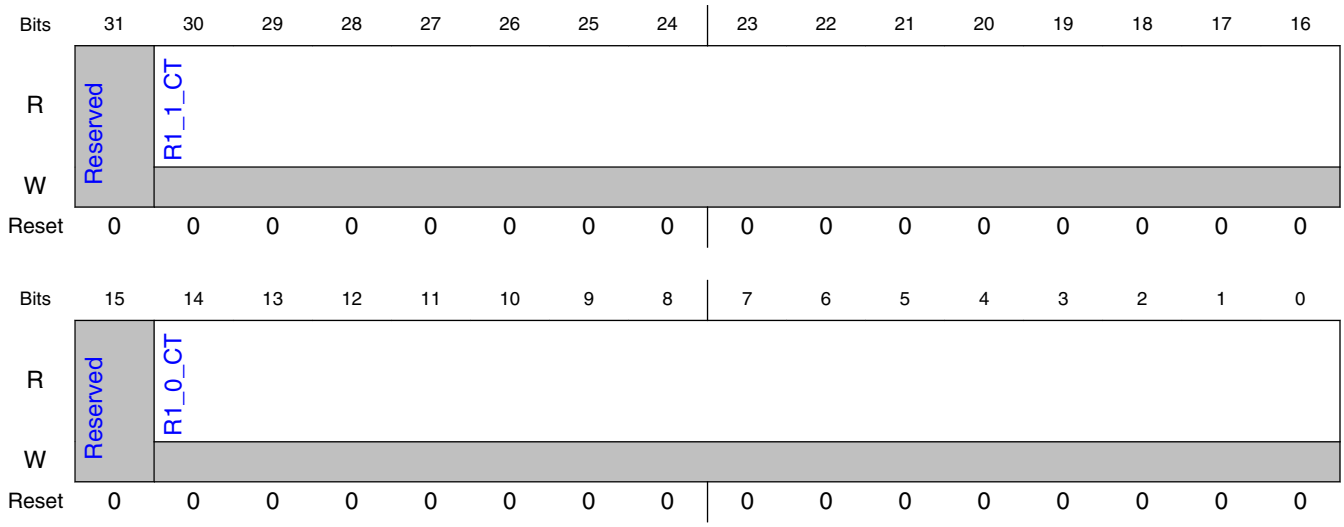
44.1.5.16.1 Offset

Register	Offset	Description
SCR1C	24h	Accessible at this address when MCTL[PRGM] = 0]

44.1.5.16.2 Function

The 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 SCR1L[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 SCR1L if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR1C readback register, as described here.

44.1.5.16.3 Diagram



44.1.5.16.4 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 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 MCTL[PRGM] = 0.

44.1.5.17 Statistical Check Run Length 2 Count Register (SCR2C)

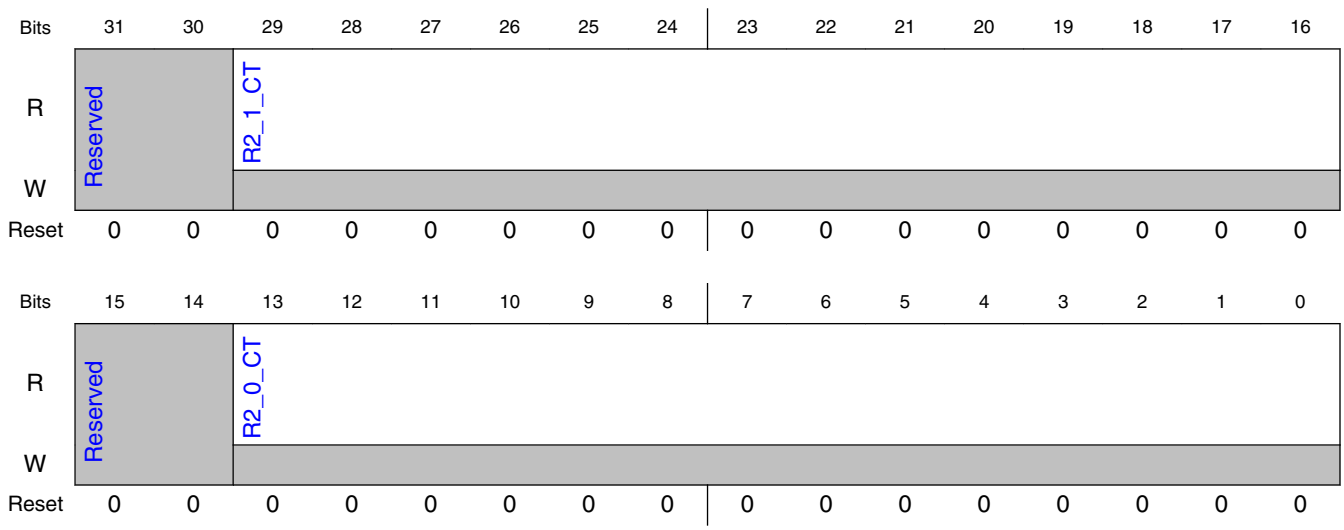
44.1.5.17.1 Offset

Register	Offset	Description
SCR2C	28h	Accessible at this address when MCTL[PRGM] = 0]

44.1.5.17.2 Function

The 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 SCR2L[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 SCR2L if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR2C readback register, as described here.

44.1.5.17.3 Diagram



44.1.5.17.4 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 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 MCTL[PRGM] = 0.

44.1.5.18 Statistical Check Run Length 2 Limit Register (SCR2L)

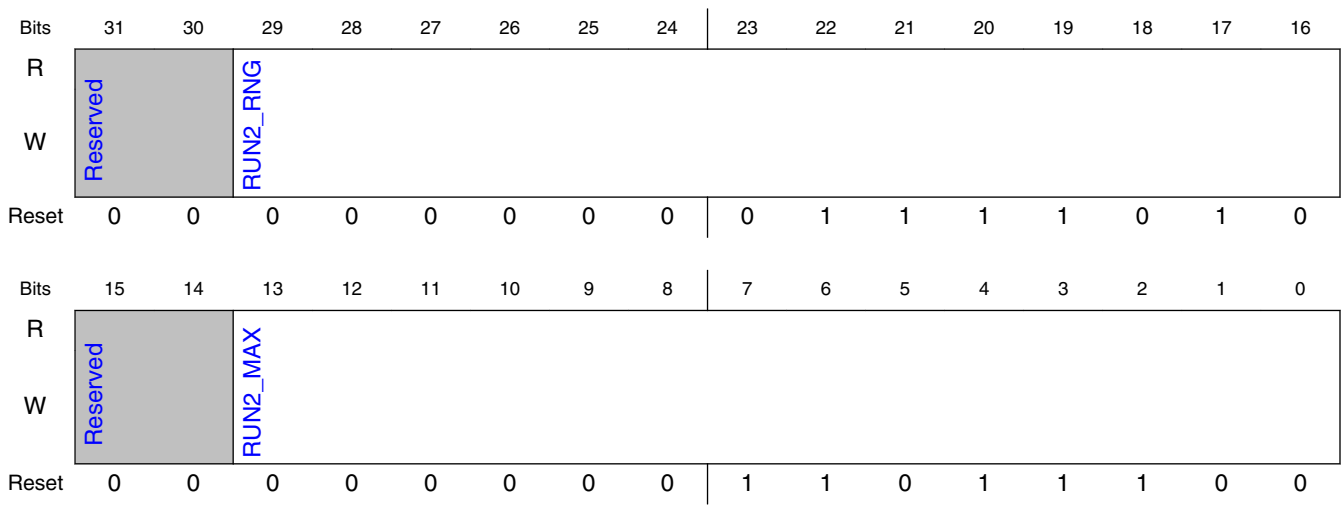
44.1.5.18.1 Offset

Register	Offset	Description
SCR2L	28h	Accessible at this address when MCTL[PRGM] = 1]

44.1.5.18.2 Function

The 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 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 SCR2L only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as SCR2C readback register.

44.1.5.18.3 Diagram



44.1.5.18.4 Fields

Field	Function
31-30	Reserved. Always 0.

Table continues on the next page...

Field	Function
—	
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 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 MCTL[RST_DEF] bit to 1.

44.1.5.19 Statistical Check Run Length 3 Limit Register (SCR3L)

44.1.5.19.1 Offset

Register	Offset	Description
SCR3L	2Ch	Accessible at this address when MCTL[PRGM] = 1]

44.1.5.19.2 Function

The 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 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 SCR3L only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as SCR3C readback register.

44.1.5.19.3 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

44.1.5.19.4 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 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 MCTL[RST_DEF] bit to 1.

44.1.5.20 Statistical Check Run Length 3 Count Register (SCR3C)

44.1.5.20.1 Offset

Register	Offset	Description
SCR3C	2Ch	Accessible at this address when MCTL[PRGM] = 0]

44.1.5.20.2 Function

The 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 SCR3L[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 SCR3L if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR3C readback register, as described here.

44.1.5.20.3 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

44.1.5.20.4 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 MCTL[PRGM] = 0.
15-13 —	Reserved. Always 0.
12-0 R3_0_CT	Runs of Zeroes, Length 3 Count. Reads the final Runs of Zeroes, length 3 count after entropy generation. Requires MCTL[PRGM] = 0.

44.1.5.21 Statistical Check Run Length 4 Count Register (SCR4C)

44.1.5.21.1 Offset

Register	Offset	Description
SCR4C	30h	Accessible at this address when MCTL[PRGM] = 0]

44.1.5.21.2 Function

The 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 SCR4L[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 SCR4L if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR4C readback register, as described here.

44.1.5.21.3 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

44.1.5.21.4 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 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 MCTL[PRGM] = 0.

44.1.5.22 Statistical Check Run Length 4 Limit Register (SCR4L)

44.1.5.22.1 Offset

Register	Offset	Description
SCR4L	30h	Accessible at this address when MCTL[PRGM] = 1]

44.1.5.22.2 Function

The 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 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 SCR4L only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as SCR4C readback register.

44.1.5.22.3 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

44.1.5.22.4 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 MCTL[RST_DEF] bit to 1.
15-12 —	Reserved. Always 0.

Table continues on the next page...

Field	Function
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 MCTL[RST_DEF] bit to 1.

44.1.5.23 Statistical Check Run Length 5 Count Register (SCR5C)

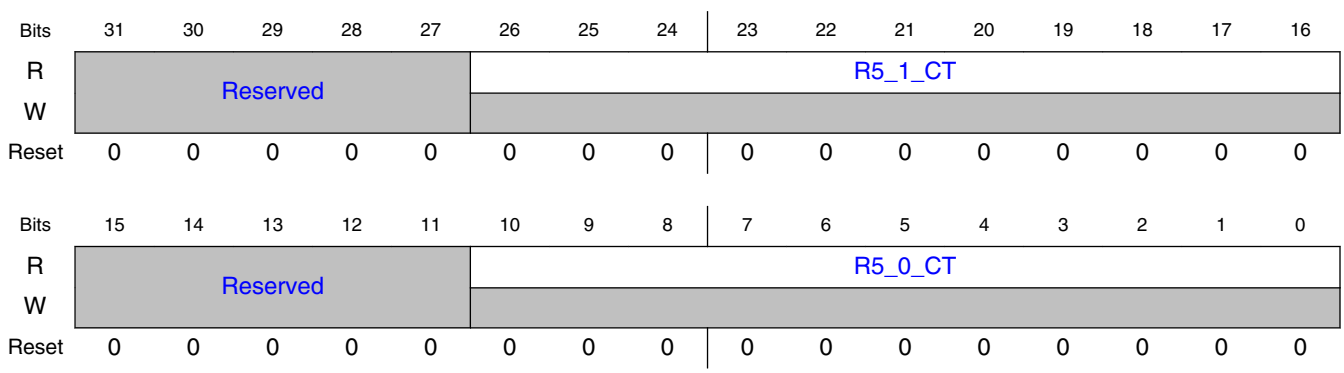
44.1.5.23.1 Offset

Register	Offset	Description
SCR5C	34h	Accessible at this address when MCTL[PRGM] = 0]

44.1.5.23.2 Function

The 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 SCR5L[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 SCR5L if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR5C readback register, as described here.

44.1.5.23.3 Diagram



44.1.5.23.4 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 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 MCTL[PRGM] = 0.

44.1.5.24 Statistical Check Run Length 5 Limit Register (SCR5L)

44.1.5.24.1 Offset

Register	Offset	Description
SCR5L	34h	Accessible at this address when MCTL[PRGM] = 1]

44.1.5.24.2 Function

The 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 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 SCR5L only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as SCR5C readback register.

44.1.5.24.3 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

44.1.5.24.4 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 MCTL[RST_DEF] bit to 1.
15-11 —	Reserved. Always 0.
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 MCTL[RST_DEF] bit to 1.

44.1.5.25 Statistical Check Run Length 6+ Count Register (SCR6PC)

44.1.5.25.1 Offset

Register	Offset	Description
SCR6PC	38h	Accessible at this address when MCTL[PRGM] = 0]

44.1.5.25.2 Function

The 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 SCR6PL[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 SCR6PL if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR6PC readback register, as described here.

44.1.5.25.3 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

44.1.5.25.4 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 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 MCTL[PRGM] = 0.

44.1.5.26 Statistical Check Run Length 6+ Limit Register (SCR6PL)

44.1.5.26.1 Offset

Register	Offset	Description
SCR6PL	38h	Accessible at this address when MCTL[PRGM] = 1]

44.1.5.26.2 Function

The 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 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 SCR6PL only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR6PC readback register.

44.1.5.26.3 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

44.1.5.26.4 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 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 MCTL[RST_DEF] bit to 1.

44.1.5.27 Status Register (STATUS)

44.1.5.27.1 Offset

Register	Offset
STATUS	3Ch

44.1.5.27.2 Function

Various statistical tests are run as a normal part of the TRNG's entropy generation process. The least-significant 16 bits of the 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 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 SCMISC register.

All 0s will be returned if this register address is read while the RNG is in Program Mode (see PRGM field in MCTL register. If this register is read while the RNG is in Run Mode the value returned will be formatted as follows.

44.1.5.27.3 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

44.1.5.27.4 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 TFMB	Test Fail, Mono Bit. If TFMB=1, the Mono Bit Test has failed.
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.

44.1.5.28 Entropy Read Register (ENT0 - ENT15)

44.1.5.28.1 Offset

For a = 0 to 15:

Register	Offset	Description
ENTa	40h + (a × 4h)	Word a

44.1.5.28.2 Function

The RNG TRNG can be programmed to generate an entropy value that is readable via the SkyBlue bus. To do this, set the MCTL[TRNG_ACC] bit to 1. Once the entropy value has been generated, the MCTL[ENT_VAL] bit will be set to 1. At this point, ENT0 through ENT15 may be read to retrieve the 512-bit entropy value. Note that once ENT15 is read, the entropy value will be cleared and a new value will begin generation, so it is important that ENT15 be read last. These registers are readable only when MCTL[PRGM] = 0 (Run Mode) , MCTL[TRNG_ACC] = 1 (TRNG access mode) and MCTL[ENT_VAL] = 1. After at most one (1) bus clock cycle of reading a valid ENT15 register value, reading any ENT0 through ENT15 register would return zeros.

44.1.5.28.3 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

44.1.5.28.4 Fields

Field	Function
31-0 ENT	Entropy Value. Will be non-zero only if MCTL[PRGM] = 0 Run Mode) and 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.

44.1.5.29 Statistical Check Poker Count 1 and 0 Register (PKRCNT10)

44.1.5.29.1 Offset

Register	Offset
PKRCNT10	80h

44.1.5.29.2 Function

The 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 MCTL[PRGM] is 0, otherwise zeroes will be read.

44.1.5.29.3 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

44.1.5.29.4 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 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 MCTL[PRGM] = 0.

44.1.5.30 Statistical Check Poker Count 3 and 2 Register (PKRCNT32)

44.1.5.30.1 Offset

Register	Offset
PKRCNT32	84h

44.1.5.30.2 Function

The 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 MCTL[PRGM] is 0, otherwise zeroes will be read.

44.1.5.30.3 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

44.1.5.30.4 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 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 MCTL[PRGM] = 0.

44.1.5.31 Statistical Check Poker Count 5 and 4 Register (PKRCNT54)

44.1.5.31.1 Offset

Register	Offset
PKRCNT54	88h

44.1.5.31.2 Function

The 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 MCTL[PRGM] is 0, otherwise zeroes will be read.

44.1.5.31.3 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

44.1.5.31.4 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 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 MCTL[PRGM] = 0.

44.1.5.32 Statistical Check Poker Count 7 and 6 Register (PKRCNT76)

44.1.5.32.1 Offset

Register	Offset
PKRCNT76	8Ch

44.1.5.32.2 Function

The 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 MCTL[PRGM] is 0, otherwise zeroes will be read.

44.1.5.32.3 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

44.1.5.32.4 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 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 MCTL[PRGM] = 0.

44.1.5.33 Statistical Check Poker Count 9 and 8 Register (PKRCNT98)

44.1.5.33.1 Offset

Register	Offset
PKRCNT98	90h

44.1.5.33.2 Function

The 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 MCTL[PRGM] is 0, otherwise zeroes will be read.

44.1.5.33.3 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

44.1.5.33.4 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 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 MCTL[PRGM] = 0.

44.1.5.34 Statistical Check Poker Count B and A Register (PKRCNTBA)

44.1.5.34.1 Offset

Register	Offset
PKRCNTBA	94h

44.1.5.34.2 Function

The 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 MCTL[PRGM] is 0, otherwise zeroes will be read.

44.1.5.34.3 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

44.1.5.34.4 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 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 MCTL[PRGM] = 0.

44.1.5.35 Statistical Check Poker Count D and C Register (PKRCNTDC)

44.1.5.35.1 Offset

Register	Offset
PKRCNTDC	98h

44.1.5.35.2 Function

The 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 MCTL[PRGM] is 0, otherwise zeroes will be read.

44.1.5.35.3 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

44.1.5.35.4 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 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 MCTL[PRGM] = 0.

44.1.5.36 Statistical Check Poker Count F and E Register (PKRCNTFE)

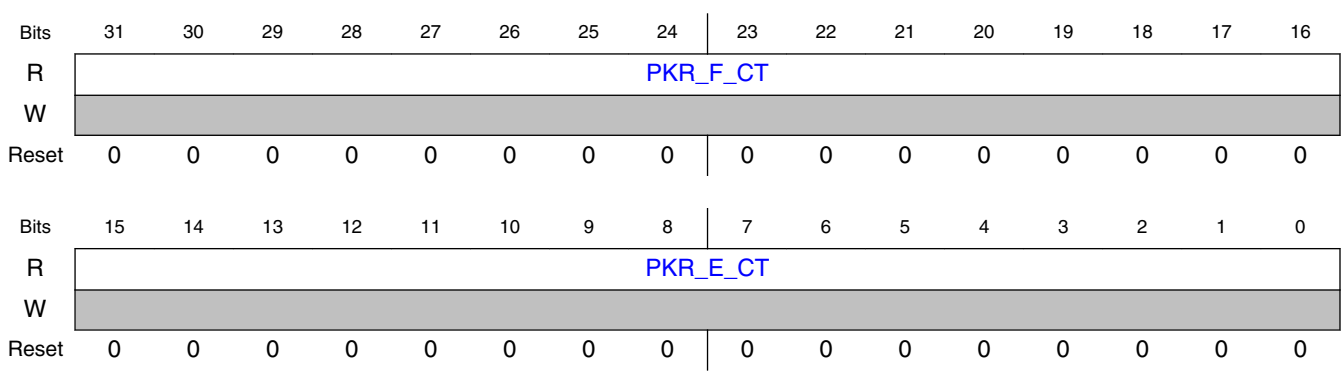
44.1.5.36.1 Offset

Register	Offset
PKRCNTFE	9Ch

44.1.5.36.2 Function

The 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 MCTL[PRGM] is 0, otherwise zeroes will be read.

44.1.5.36.3 Diagram



44.1.5.36.4 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 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 MCTL[PRGM] = 0.

44.1.5.37 Security Configuration Register (SEC_CFG)

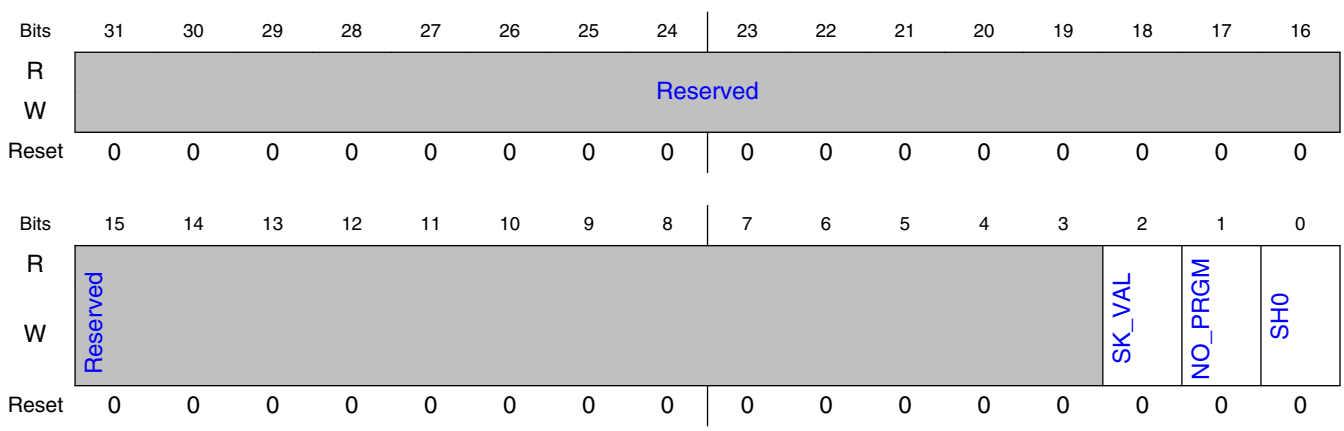
44.1.5.37.1 Offset

Register	Offset
SEC_CFG	B0h

44.1.5.37.2 Function

The 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.

44.1.5.37.3 Diagram



44.1.5.37.4 Fields

Field	Function
31-3 —	Reserved.
2 SK_VAL	Reserved. DRNG-specific, not applicable to this version. 0b - See DRNG version. 1b - 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. 0b - Programability of registers controlled only by the Miscellaneous Control Register's access mode bit. 1b - Overides Miscellaneous Control Register access mode and prevents TRNG register programming.
0 SH0	Reserved. DRNG specific, not applicable to this version. 0b - See DRNG version. 1b - See DRNG version.

44.1.5.38 Interrupt Control Register (INT_CTRL)

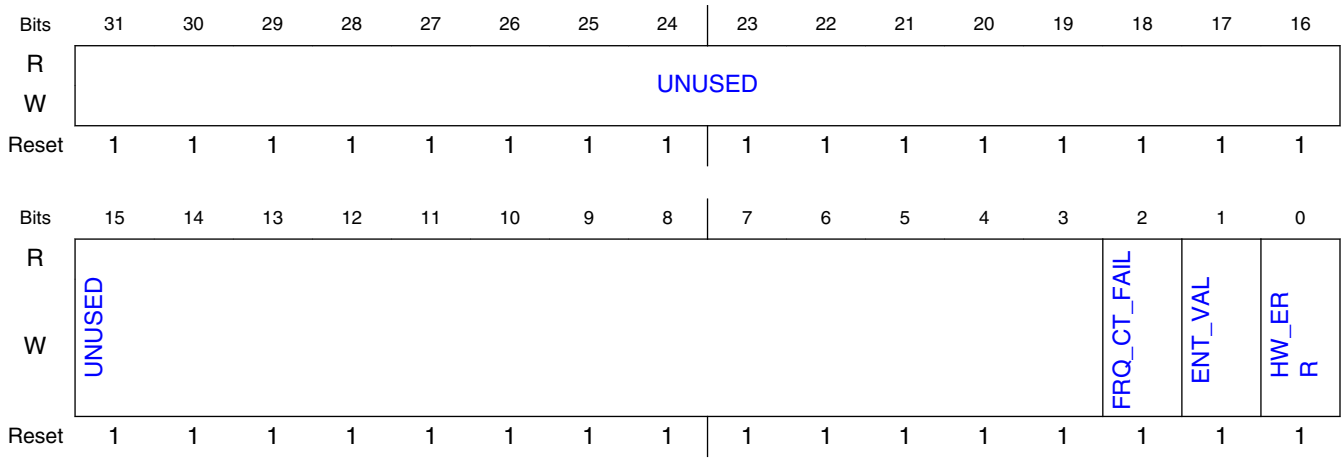
44.1.5.38.1 Offset

Register	Offset
INT_CTRL	B4h

44.1.5.38.2 Function

The 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 INT_STATUS register description above. Each interrupt can be cleared by de-asserting the corresponding bit in the 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 MCTL register.

44.1.5.38.3 Diagram



44.1.5.38.4 Fields

Field	Function
31-3 UNUSED	Reserved but writeable.
2 FRQ_CT_FAIL	Same behavior as bit 0 above. 0b - Same behavior as bit 0 above. 1b - Same behavior as bit 0 above.
1 ENT_VAL	Same behavior as bit 0 above. 0b - Same behavior as bit 0 above. 1b - Same behavior as bit 0 above.
0 HW_ERR	Bit position that can be cleared if corresponding bit of INT_STATUS has been asserted. 0b - Corresponding bit of INT_STATUS cleared. 1b - Corresponding bit of INT_STATUS active.

44.1.5.39 Mask Register (INT_MASK)

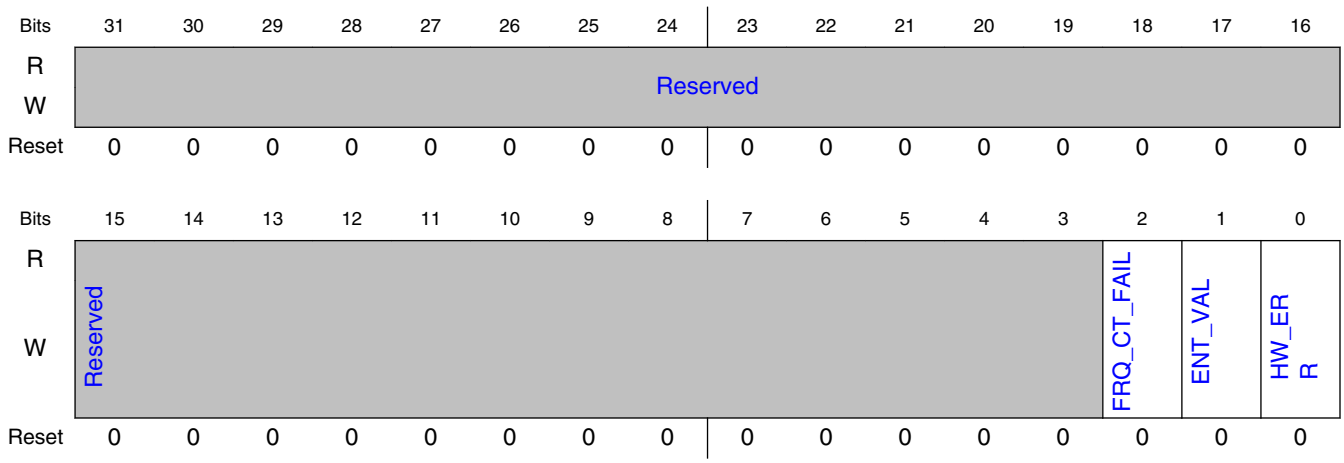
44.1.5.39.1 Offset

Register	Offset
INT_MASK	B8h

44.1.5.39.2 Function

The 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 INT_STATUS register description above. Each interrupt can be masked/disabled by de-asserting the corresponding bit in the 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 MCTL register.

44.1.5.39.3 Diagram



44.1.5.39.4 Fields

Field	Function
31-3 —	Reserved.
2 FRQ_CT_FAIL	Same behavior as bit 0 above. 0b - Same behavior as bit 0 above. 1b - Same behavior as bit 0 above.
1 ENT_VAL	Same behavior as bit 0 above. 0b - Same behavior as bit 0 above. 1b - Same behavior as bit 0 above.
0 HW_ERR	Bit position that can be cleared if corresponding bit of INT_STATUS has been asserted. 0b - Corresponding interrupt of INT_STATUS is masked. 1b - Corresponding bit of INT_STATUS is active.

44.1.5.40 Interrupt Status Register (INT_STATUS)

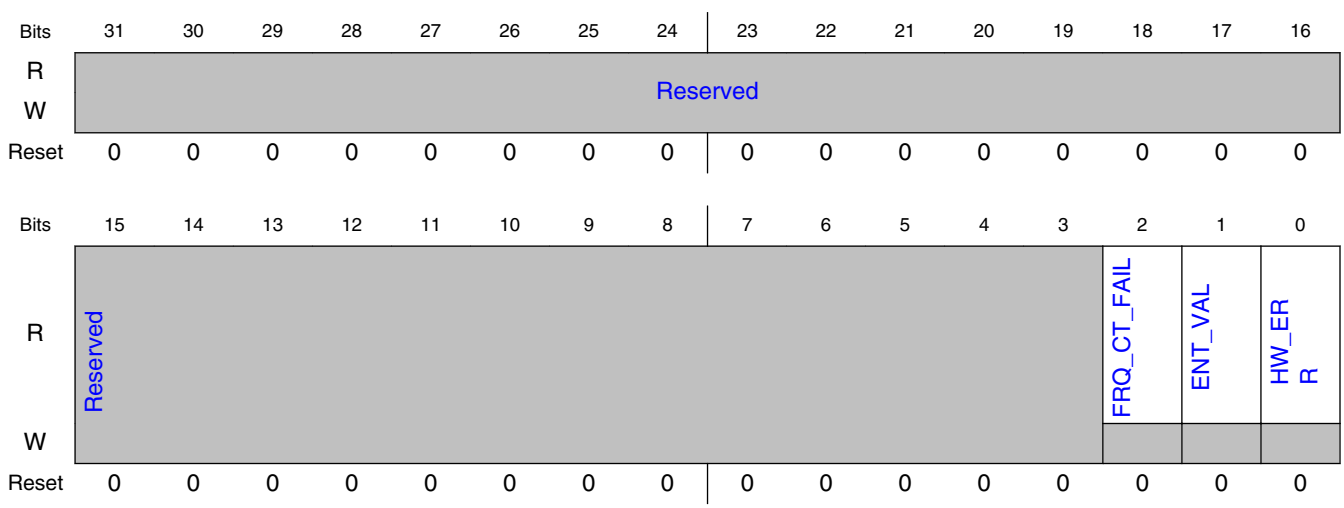
44.1.5.40.1 Offset

Register	Offset
INT_STATUS	BCh

44.1.5.40.2 Function

The 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 INT_STATUS register. Each interrupt can be temporarily cleared by de-asserting the corresponding bit in the INT_CTRL register. To mask the interrupts, clear the corresponding bits in the INT_MASK register. The description of each of the 3 interrupts is defined in the Block Guide under the MCTL register description. Even if the interrupt is cleared or masked, interrupt status information can be read from the MCTL register.

44.1.5.40.3 Diagram



44.1.5.40.4 Fields

Field	Function
31-3	Reserved.

Table continues on the next page...

Field	Function
—	
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 FRQMAX and/or FRQMIN registers, or a hardware failure in the ring oscillator. 0b - No hardware nor self test frequency errors. 1b - 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 ENT15 is read. (ENT0 through ENT14 should be read before reading ENT15). 0b - Busy generation entropy. Any value read is invalid. 1b - 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. 0b - no error 1b - error detected.

44.1.5.41 Version ID Register (MS) (VID1)

44.1.5.41.1 Offset

Register	Offset
VID1	F0h

44.1.5.41.2 Function

The Version ID Register is a read only register used to identify the version of the TRNG in use. This register as well as VID2 should both be read to verify the expected version.

44.1.5.41.3 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	0	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

44.1.5.41.4 Fields

Field	Function
31-16 IP_ID	Shows the IP ID. 0000000000110000b - ID for TRNG.
15-8 MAJ_REV	Shows the IP's Major revision of the TRNG. 00000001b - Major revision number for TRNG.
7-0 MIN_REV	Shows the IP's Minor revision of the TRNG. 00000000b - Minor revision number for TRNG.

44.1.5.42 Version ID Register (LS) (VID2)

44.1.5.42.1 Offset

Register	Offset
VID2	F4h

44.1.5.42.2 Function

The Version ID Register LSB is a read only register used to identify the architecture of the TRNG in use. This register as well as VID1 should both be read to verify the expected version.

44.1.5.42.3 Diagram

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

44.1.5.42.4 Fields

Field	Function
31-24 ERA	Shows the compile options for the TRNG. 00000000b - COMPILE_OPT for TRNG.
23-16 INTG_OPT	Shows the integration options for the TRNG. 00000000b - INTG_OPT for TRNG.
15-8 ECO_REV	Shows the IP's ECO revision of the TRNG. 00000000b - TRNG_ECO_REV for TRNG.
7-0 CONFIG_OPT	Shows the IP's Configuaration options for the TRNG. 00000000b - TRNG_CONFIG_OPT for TRNG.

Chapter 45
2.4 GHz Multi-Protocol Radio

45.1 Introduction

This document describes the 2.1 generation 2.4 GHz Multi-Protocol Radio capable of supporting the modes described below in the 2.4 GHz ISM band.

- IEEE Std. 802.15.1 v5.0 Bluetooth Low Energy (BLE) single-mode device operation
- Generic FSK at 1Mbps, 500Kbps, or 250Kbps

The Radio is comprised of a Constant-Envelope Transmitter and a Quadrature Zero-IF Receiver.

The Radio block diagram is provided for illustration of the internal organization of the radio.

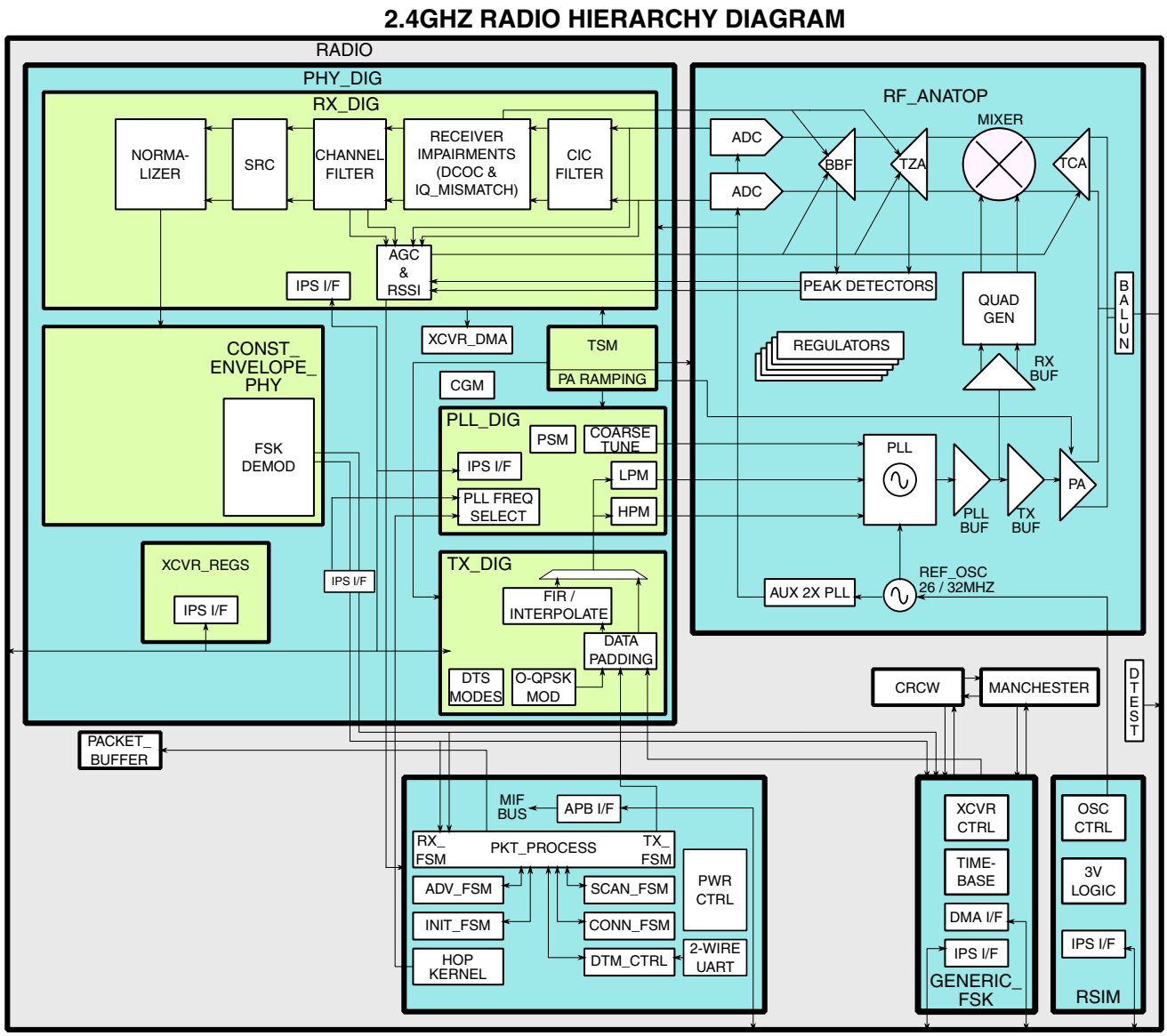


Figure 45-1. Radio Block diagram

45.2 Radio System Integration Module

45.2.1 Introduction

45.2.1.1 About the Radio System Integration Module

The Radio System Integration Module (RSIM) provides support for the Radio specific System Integration requirements.

An overview of the RSIM can be seen in the block diagram, detailed descriptions are in the sections that follow.

45.2.1.2 Block diagram

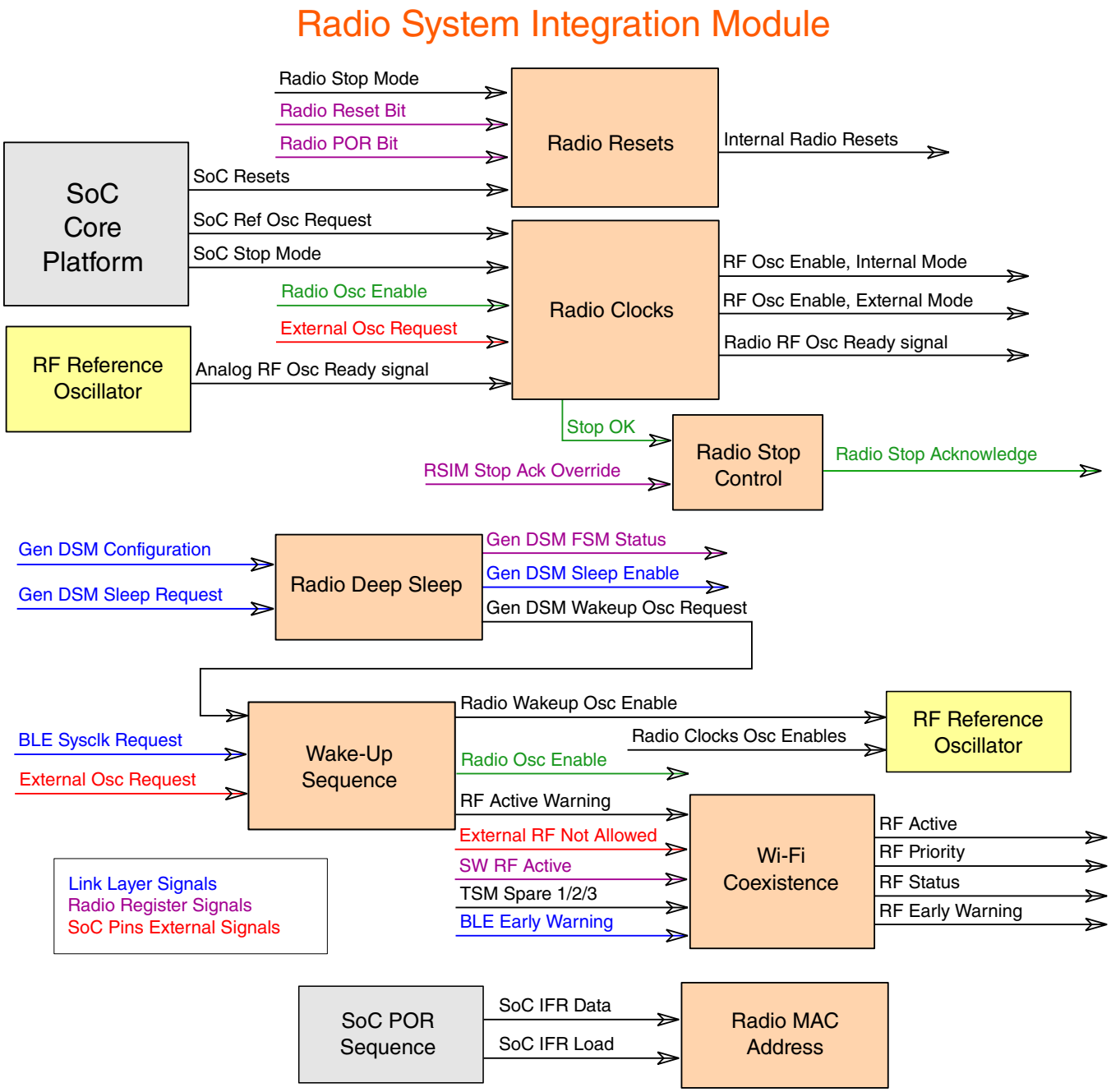


Figure 45-2. Block diagram

45.2.2 Radio Clocks

The Radio uses the RF Analog Reference Oscillator or an equivalently precise external clock for all Radio Operations. This RF Osc is also available for the SoC to use as it's clock source. There is also a method that allows the SoC to supply a less precise clock to the Radio that can be used to configure the Radio registers before the Radio is enabled.

45.2.2.1 Clock Diagram

The Radio Clock Tree is illustrated in the following diagram.

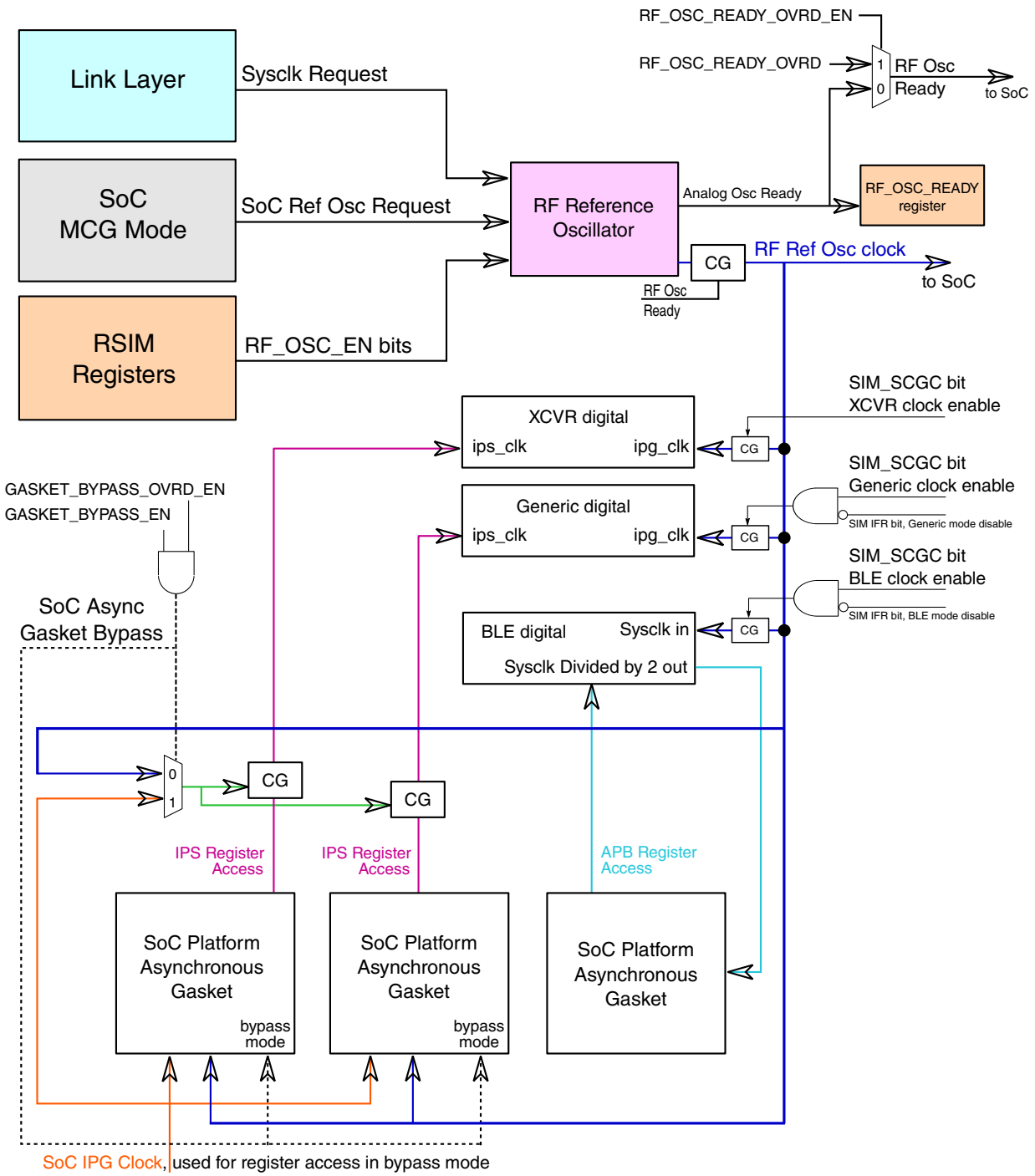


Figure 45-3. Radio Clocking Configuration

45.2.2.2 RF Osc Enable

The RF Analog Reference Oscillator must be enabled and ready before it can be used as a clock source. There are four methods of enabling the RF Osc as listed here:

1. The SoC can request the RF Osc.

2. One of the Radio Link Layers can request the RF Osc, either in Run mode or in LLS mode using a Deep Sleep State Machine.
3. The external pin can be asserted to request RF Osc be enabled.
4. The RF_OSC_EN bit can be set, more details in the table below.

RF_OSC_EN	Radio Run Mode Functionality	Radio Stop Modes Functionality
0	RF Ref Osc will be Off unless it is turned On by the SoC, by a Radio link layer or by an External Pin request.	Warning: Radio will not go into Stop mode while the RF Ref Osc is being used by the SoC, or by a Radio link layer unless the Link Layer is using a Deep Sleep State Machine. RF Ref Osc will be Off unless there is an external pin request.
1	The RF Ref Osc will be On.	Radio will not go into Stop mode while RF_OSC_EN is set

45.2.2.3 RF Osc Ready

The RF Analog Reference Oscillator has a programmable count time that elapses before the RF_OSC_READY signal is asserted. This count time is set using the BB_XTAL_READY_COUNT_SEL register as shown in the table below:

BB_XTAL_READY_COUNT_SEL	Count Value	Count Time, us (32MHz Ref Osc)
0	1024	32
1	2048	64
2	4096	128
3	8192	256

The current status of RF_OSC_READY signal from the RF Analog can be read using the RF_OSC_READY bit.

The RF_OSC_READY signal from the RF Analog can be overridden using the RF_OSC_READY_OVRD_EN and RF_OSC_READY_OVRD bits. The status bit will continue to show the RF Analog status but everywhere else will use the overridden value.

45.2.2.4 RF Osc Bypass

The RF_OSC_BYPASS_EN 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 RF EXTAL clock as the RF Ref Osc clock. Note that the RF Ref Osc Ready signal functions normally in RF OSC Bypass mode, unless overridden with the RF_OSC_READY_OVRD_EN bit.

45.2.2.5 RF Osc Debug Visibility

The RF_OSC may be made visible externally for debug purposes through the following code sequence:

```
// Set XO on CLK_out
RSIM->ANA_TRIM &= ~RSIM_ANA_TRIM_BB_XTAL_SPARE_MASK;
// Set SPARE to 4
XCVR_ANA->RX_AUXPLL &= ~XCVR_ANALOG_RX_AUXPLL_SPARE_MASK;
XCVR_ANA->RX_AUXPLL |= XCVR_ANALOG_RX_AUXPLL_SPARE(4);

RSIM->RF_OSC_CTRL |= RSIM_RF_OSC_CTRL_RADIO_EXT_OSC_OVRD_EN_MASK |
                     RSIM_RF_OSC_CTRL_RADIO_EXT_OSC_OVRD_MASK |
                     RSIM_RF_OSC_CTRL_RADIO_EXT_OSC_RF_EN_SEL_MASK;

RSIM->ANA_TEST |= RSIM_ANA_TEST_XTAL_OUT_BUF_EN_MASK;
```

45.2.2.6 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.

45.2.2.6.1 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.

Software must use the RSIM_RUN_REQUEST bit to put the Radio in RRun Mode and then enable the RF_OSC_EN bit to enable the RF Ref Osc. 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.

45.2.2.6.2 XCVR and Link Layer Registers Access

The Radio Registers for the XCVR and the Link Layers 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 behavior 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. Software must use the RSIM_RUN_REQUEST bit to put the Radio in RRun Mode and then enable the RF_OSC_EN bit to enable the RF Ref Osc. 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.

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 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.

45.2.2.6.3 Asynchronous Gasket Bypass

The intent of Bypass Mode is to allow software to configure the Radio XCVR and Link Layer 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 Link Layer registers.

Gasket Bypass Override Enable	Gasket Bypass Override	XCVR and Link Layer Register Clock
1	0	RF Ref Osc Clock

Table continues on the next page...

Gasket Bypass Override Enable	Gasket Bypass Override	XCVR and Link Layer Register Clock
1	1	SoC IPG Clock
0	x	RF Ref Osc Clock

More details are available in the Radio Clocks diagram, the Radio Gasket Bypass section, and in the RSIM Gasket Bypass Registers descriptions.

45.2.2.7 External Osc Request

The RF Analog Reference Oscillator has an External Osc mode in which the RF Osc Analog circuits can be enabled and the RF Osc can be ready and either presented externally on an SoC pin, or not.

45.2.2.7.1 SoC XTAL_OUT Pin Enable

The RF Analog Driver for the SoC XTAL_OUT pin has to be enabled using the XTAL_OUT_BUF_EN bit in the RSIM ANA_TEST register.

Warning: If this bit is not set then no RF Osc signal will be presented on the SoC Pin.

45.2.2.7.2 External Osc Request Register Bits

The RSIM has several bits that are used to select the External Osc mode configuration as shown in the table below:

Table 45-1. External Osc Mode Configuration Bits

RSIM Register Field	Function	Setting	Description
BLOCK_EXT_OSC_PWR_REQ	Block External Requests for RF OSC from starting a Radio Power Wakeup Sequence	0	Configuring the RF Osc in External Osc mode causes the Radio to request Run Regulation mode.
		1	Radio will not request Run Regulation mode when the RF Osc is configured in External Osc mode and the Radio is otherwise in a Stop mode.
RADIO_EXT_OSC_RF_EN_SEL	Radio External Request for RF OSC Select	0	SoC pin connected to EXT_OSC_RF_EN_0 is used to configure the RF Osc in External Osc mode.

Table continues on the next page...

Table 45-1. External Osc Mode Configuration Bits (continued)

		1	SoC pin connected to EXT_OSC_RF_EN_1 is used to configure the RF Osc in External Osc mode.
RADIO_EXT_OSC_OVRD	Radio External Request for RF OSC Override	0	If the RADIO_EXT_OSC_OVRD_EN bit is set, then the RF Osc will not be configured in External Osc mode.
		1	If the RADIO_EXT_OSC_OVRD_EN bit is set, then the RF Osc will be configured in External Osc mode.
RADIO_EXT_OSC_OVRD_EN	Radio External Request for RF OSC Override Enable	0	SoC external pin requests will control the RF Osc External Osc mode configuration.
		1	The RADIO_EXT_OSC_OVRD bit will control the RF Osc External Osc mode configuration and any external pin request will be ignored.

45.2.2.8 SoC RF Osc Pins

The SoC Pins related to the RF Osc are shown in the table below.

Radio Pin Name	Pin Direction	SoC Pin Name	Pin Description
RF_INT_OSC_EN	I	RF_RFOSC_EN	External request to turn on the Radio's internal RF Osc. The Osc will be present on the XTAL_OUT pin if the RF Osc XTAL_OUT buffer is enabled by writing the Radio XTAL_OUT_BUF_EN bit in the RSIM ANA_TEST register.
RF_EXT_OSC_EN	O	RF_EXT_OSC_EN	Internal request to turn on an External Osc for use by the internal Radio, this can also be a request from the SoC if it is using the RF Osc as it's clock.

45.2.3 Radio Stop Modes

The Radio hardware automatically exits and reenters Stop Mode as part of its interaction with the Deep Sleep State Machines.

45.2.3.1 RF Analog Isolation

The RSIM controls the isolation state of the RF Analog in Stop Modes, the detailed signaling description is shown in the following diagram:

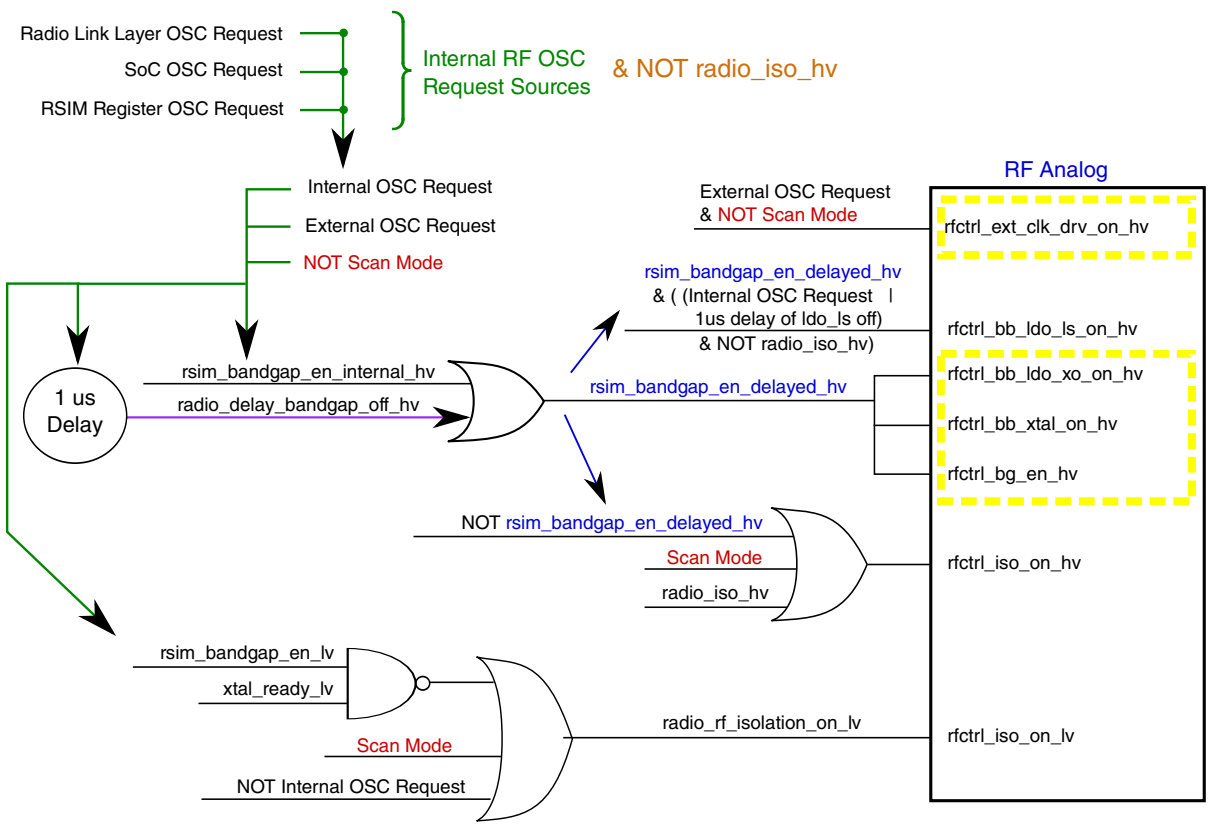


Figure 45-4. RF Analog Isolation Diagram

45.2.3.2 RAM Retention

The Radio has two internal RAM blocks that are allowed to go into a low power state when they are not being used by a link layer. If the RADIO_RAM_ACCESS_OVRD_EN and RADIO_RAM_ACCESS_OVRD bits are set, then the RAMs are kept in an active power state to allow software to access them.

WARNING: Software should not cause the RAMs to lose power if the Radio Link Layers are using the Deep Sleep State Machines and are configured to use the RAM to store data.

45.2.3.3 Radio Idle Mode

When the Radio is in a low power Stop Mode, there is a software method that allows the RF Analog Reference Oscillator to be kept enabled in a low power state and thereby warmed up and ready for a faster transition from Stop to Run mode.

In order to configure for this Idle Mode software should write these bits before going into Stop:

- RSIM_ANA_TEST_XTAL_OUT_BUF_EN = 0, Turn off the External Pin Drive Buffer so that no Osc Clock will be present.
- RSIM_SW_CONFIG_BLOCK_EXT_OSC_PWR_REQ = 1, Radio will not request full Run Regulation mode when EXT_OSC is requested.
- RSIM_RF_OSC_CTRL_RADIO_EXT_OSC_OVRD_EN = 1, Enable the override bit.
- RSIM_RF_OSC_CTRL_RADIO_EXT_OSC_OVRD = 1, This is the software request to enable the Osc in Idle Mode.

45.2.4 Radio Deep Sleep

The RSIM has a Deep Sleep State Machine to support Generic Deep Sleep mode.

Note that the BLE Link Layer has it's own Deep Sleep FSM, please refer to the BLE Link Layer chapters for additional information.

This section covers the portions of the Deep Sleep Hardware contained in the RSIM.

45.2.4.1 Deep Sleep Entry

The RSIM Deep Sleep State Machine will enter sleep automatically based on hardware requests from the link layer or the time as counted in the DSM_TIMER. Please refer to the Generic DSM section for more details.

45.2.4.2 Deep Sleep Wakeup

The RSIM Deep Sleep State Machine will automatically wakeup based upon the time as counted in the DSM_TIMER.

The DSM_TIMER is enabled using the DSM_TIMER_EN bit, and the timer can be cleared using the DSM_TIMER_CLR bit.

45.2.4.2.1 Wakeup Sequence Settings

Software must configure the Deep Sleep Wakeup Sequence using these RSIM register bits. The Wakeup Sequence will be the same for all of the Radio Deep Sleep Wakeups.

The required settings and the method of calculating them are shown in the following table.

Table 45-2. Wakeup Sequence Settings

RSIM Register Field	Function	Description
COARSE_DELAY	Deep Sleep Wakeup Coarse Delay Time	This is the first delay after the beginning of the Wake-Up sequence, the range is 0-15 and the resulting delay is 1.953 ms x COARSE_DELAY + 30.52 us, with a minimum delay of 30.52 us.
FINE_DELAY	Deep Sleep Wakeup Fine Delay Time	This is the second delay after the beginning of the Wake-Up sequence, the range is 0-63 and the resulting delay is 122.07 us x FINE_DELAY + 30.52 us, with a minimum delay of 30.52 us. This is when the RF OSC is enabled and so the Radio Analog Power must be stable at this milestone.
ACTIVE_WARNING	Deep Sleep Wakeup RF Active Warning Time	This is the delay after RF OSC has been enabled, the range is 0-63 and the resulting delay is 122.07 us x ACTIVE_WARNING + 30.52 us, with a minimum delay of 30.52 us. This is typically when the RF Osc should be stable.

45.2.4.3 Deep Sleep Timer

When the Deep Sleep Timer is enabled, using the DSM_TIMER_EN bit, the DSM_TIMER register represents the 24-bit internal counter that counts the 32 kHz clock cycles. The DSM_TIMER register is cleared while the DSM_TIMER_CLR bit is set.

The Deep Sleep State Machines use the DSM_TIMER value to calculate sleep and wakeup times.

45.2.4.4 Deep Sleep Forced Exit

While the BLE Deep Sleep State Machine is in sleep mode software can force a wakeup using the BLE_DSM_EXIT register bit.

The State Machine will hold in its initial idle state until the BLE_DSM_EXIT register bit is cleared.

45.2.4.5 Deep Sleep State Machines

The RSIM registers have fields that are used by software to configure, control, and observe the Deep Sleep State Machines, these are described in the following tables.

Table 45-3. Generic Deep Sleep State Machine Registers

RSIM Register Field	Function	Description
GEN_FSM_STATE	GEN Deep Sleep State Machine State	This is the current State that the GEN Deep Sleep Finite State Machine is in.
DSM_GEN_READY	GEN Ready for Deep Sleep Mode	This is the SLEEP_READY state in the Deep Sleep Module State Machine, the state machine holds here until the sleep time is reached on the 32 kHz clock counter (Deep Sleep Timer).
GEN_DEEP_SLEEP_STATUS	GEN Deep Sleep Mode Status	This is the signal from the Deep Sleep Module State Machine telling the Link Layer to enter and exit Deep Sleep Mode. If this bit is set then the Link Layer is in Deep Sleep Mode.
DSM_GEN_FINISHED	GEN Deep Sleep Time Finished	This register is for debug purposes. This is the SLEEP_FINISHED state in the Deep Sleep Module State Machine, the state machine holds here until the Link Layer de-asserts the sleep_request.
GEN_WAKEUP_REQUEST_EN	Enable GEN Deep Sleep Module to initiate a Radio Wakeup	This bit allows the GEN Deep Sleep Module to initiate a Radio Wakeup.
GEN_SLEEP_REQUEST	GEN Deep Sleep Requested	From the GEN Sleep Enable register. This enables a match of GEN_SLEEP[23:0] to SLEEP_TMR[23:0]; when the match occurs Deep Sleep Mode is entered.
GEN_WAKEUP_REQ	GEN Deep Sleep Module Radio Wakeup Status	This bit shows the status of the GEN Deep Sleep Module Request for the Radio to Wakeup. If the GEN_WAKEUP_REQUEST_EN bit is set, then this signal is used by the RSIM to Wakeup the Radio if the Radio is not already awake.

Table continues on the next page...

Table 45-3. Generic Deep Sleep State Machine Registers (continued)

GEN_WAKEUP_INTERRUPT_EN	GEN Deep Sleep Module Radio Wakeup Interrupt Enable	If this bit is set, then a change from a 0 to a 1 on GEN_WAKEUP_REQ will generate an interrupt which can be used by the SoC.
GEN_WAKEUP_REQ_INT	Interrupt Flag from an GEN Deep Sleep Module Radio Wakeup	If the GEN_WAKEUP_INTERRUPT_EN bit is set, then this flag indicates that an GEN Radio Wakeup has occurred. The flag will persist until this bit is written with a 1 to clear it.
GEN_SLEEP_TIME	GEN Deep Sleep Module Sleep Time	Software determines the future time at which it would like to enter Deep Sleep Mode by reading the DSM_TIMER, computing the number of 32 kHz clock cycles remaining until the desired DSM start-time, and writing that value into this register. The value programmed into this register should be no fewer than 4 clocks greater (in the future) than the current time as read from DSM_TIMER.
GEN_WAKE_TIME	GEN Deep Sleep Module Wake Time	Software determines the future time at which it would like to exit Deep Sleep Mode by reading the DSM_TIMER, computing the number of 32 kHz clock cycles remaining until the desired DSM exit time, and writing that value into this register.

45.2.5 Wi-Fi Coexistence

Hardware and Software methods allow the Radio to coexist in the same space as a Wi-Fi transceiver IC, which shares the same 2.4GHz frequency band. The coexistence scheme designates the Wi-Fi transceiver as the master, and the Radio as the slave. The objective of the coexistence strategy is to prevent both the Wi-Fi and Radio from transmitting simultaneously; a configuration option exists to also prevent the Radio from receiving when the Wi-Fi transceiver owns the RF channel.

A very basic mechanism is used in cases where the Radio does not generate requests, but instead defers to the Wi-Fi transceiver to grant permission to use the medium. The Wi-Fi IC generates a signal, RF_NOT_ALLOWED. If this signal is asserted, then the Radio does not perform any communication. When this signal is de-asserted, then the Radio is free to perform communications. In case the signal RF_NOT_ALLOWED is de-asserted when the Radio has already initiated the transmission/reception of a packet, then the Radio must stop its activity immediately .

Inside the Radio, all protocols are potentially affected by the coexistence strategy. However, coexistence-driven aborting can be enabled or disabled for the different protocols, under software control.

45.2.5.1 Wi-Fi Coexistence Pins

The Wi-Fi Coexistence Pins are described in the following table.

Wi-Fi Coexist Function	Pin Direction	SoC Pin Name	Pin Description
RF_ACTIVE	O	RF_ACTIVE	An output which is asserted prior to any Radio event and remains asserted for the duration of the event.
RF_STATUS	O	RF_STATUS	An output which indicates when the Radio is in an RX or TX event, software can also control this signal directly.
RF_PRIORITY	O	RF_PRIORITY	An output which indicates to the external WiFi device that the Radio event is a high priority and it needs access to the 2.4GHz antenna.
RF_EARLY_WARNING	O	RF_EARLY_WARNING	BLE LL generated signal which can be used to wake an external sensor to make a measurement before a BLE event
RF_NOT_ALLOWED	I	RF_NOT_ALLOWED	External signal which causes the internal Radio to cease radio activity.
RF_TX_CONF	I	RF_TX_CONF	Signal from an external Radio which indicates the availability of the 2.4GHz antenna to the internal Radio.

45.2.5.2 RF Not Allowed Register Bits

Several register bits have been provided in the XCVR_CTRL registers to enable, control, and provide status for RF_NOT_ALLOWED events. These bits are described in the following table.

Field	R/W	Description
RF_NOT_ALLOWED_NO_TX	RW	1: Allow assertions on RF_NOT_ALLOWED to abort a TSM TX sequence 0: Don't allow such aborts on TX
RF_NOT_ALLOWED_NO_RX	RW	1: Allow assertions on RF_NOT_ALLOWED to abort a TSM RX sequence 0: Don't allow such aborts on RX
RF_NOT_ALLOWED_ASSERTED	R, Write a 1 to clear	1: An assertion on RF_NOT_ALLOWED has occurred since the last time this bit was cleared 0: No such assertion has occurred since the last time bit was cleared **Note**: this bit will become set regardless of whether aborting on RF_NOT_ALLOWED events is enabled or not
RF_NOT_ALLOWED_TX_ABORT	R, Write a 1 to clear	1: An assertion on RF_NOT_ALLOWED has aborted a TSM TX sequence since the last time this bit was cleared 0: No such TX abort has occurred since the last time bit was cleared
RF_NOT_ALLOWED_RX_ABORT	R	1: An assertion on RF_NOT_ALLOWED has aborted a TSM RX sequence since the last time this bit was cleared 0: No such RX abort has occurred since the last time bit was cleared
RF_NOT_ALLOWED	R	Reflects the raw state of the RF_NOT_ALLOWED GPIO pin

45.2.5.3 RF Active Register Bits

Several register bits have been provided in the RSIM registers to enable and control the RF_ACTIVE signaling. These bits are described in the following table.

ACTIVE_WARNING	Deep Sleep Wakeup RF Active Warning Time	This is the delay after RF OSC has been enabled, the range is 0-63 and the resulting delay is 122.07 us x ACTIVE_WARNING + 30.5 us. This is typically when the RF Osc should be stable.
WIFI_COEXIST_1	RF_ACTIVE Source	If this bit is set the RF_ACTIVE pin will be controlled by TSM Spare 1, otherwise it will be controlled by the RSIM RF Active Sequence along with any override by the SW_RF_ACTIVE_BIT.
WIFI_COEXIST_2	RF_STATUS Source	If this bit is set the RF_STATUS pin will be controlled by TSM Spare 3, otherwise it will be controlled by TSM Spare 2.

Table continues on the next page...

WIFI_COEXIST_3	RF_EARLY_WARNING Source	If this bit is set the RF_EARLY_WARNING pin will be controlled by TSM Spare 3, otherwise it will be controlled by the BLE Link Layer.
RF_ACTIVE_ENDS_WITH_TSM	RF_ACTIVE clearing mechanism	If this bit is set then the RSIM RF_ACTIVE OR source will clear when the TSM sequence ends.
SW_RF_ACTIVE_EN	Software RF_ACTIVE Control Enable	If this bit is set the SW_RF_ACTIVE_BIT acts as an OR source for the RF_ACTIVE pin.
SW_RF_ACTIVE_BIT	Software RF_ACTIVE Control Bit	If the SW_RF_ACTIVE_EN bit is set this bit acts as an OR source for the RF_ACTIVE pin.
SW_RF_ACTIVE_ENDS_WITH_TSM	Software RF_ACTIVE clearing mechanism	If this bit is set along with the SW_RF_ACTIVE_EN bit, then the Software RF_ACTIVE OR source will clear when the TSM sequence ends.
RSIM_RF_ACTIVE_OVRD	RF Active Internal Override	If the RSIM_RF_ACTIVE_OVRD_EN bit is set, then this bit controls the Internal version of RF Active instead of the Link Layers. The Internal version of RF Active whether overridden or not has a lower priority and can always be overridden by software using the SW_RF_ACTIVE_BIT or the TSM Spare 1 bit if it is selected using the WIFI_COEXIST_1 select bit.
RSIM_RF_ACTIVE_OVRD_EN	RF Active Internal Override Enable	This bit enables the RF Active Internal Override
RF_NOT_ALLOWED_OVRD	RF Not Allowed Override	If the RF_NOT_ALLOWED_OVRD_EN bit is set, then this bit controls the RF Not Allowed functionality instead of the SoC pin.
RF_NOT_ALLOWED_OVRD_EN	RF Not Allowed Override Enable	This bit enables the RF Not Allowed Override

45.2.6 RF Early Warning

The BLE Link Layer can be programmed to provide an Early Warning signal that can assert up to 38.75 ms before related the Radio Operation begins. One use of this signal is to wakeup an external sensor so that it can collect and prepare any required data that needs to be available at the time of the Radio Operation.

Please refer to the BLE RF_ACTIVE_PERIOD register description for more details.

Note that this signal can be reconfigured as part of the Wi-Fi Coexistence Method as described in that section.

45.2.7 Radio Interrupts

The Radio can generate two asynchronous interrupts and one asynchronous MCU wakeup signal.

The interrupt configurations are illustrated in the diagram below and detailed in the following sections.

45.2.7.1 Interrupts Diagram

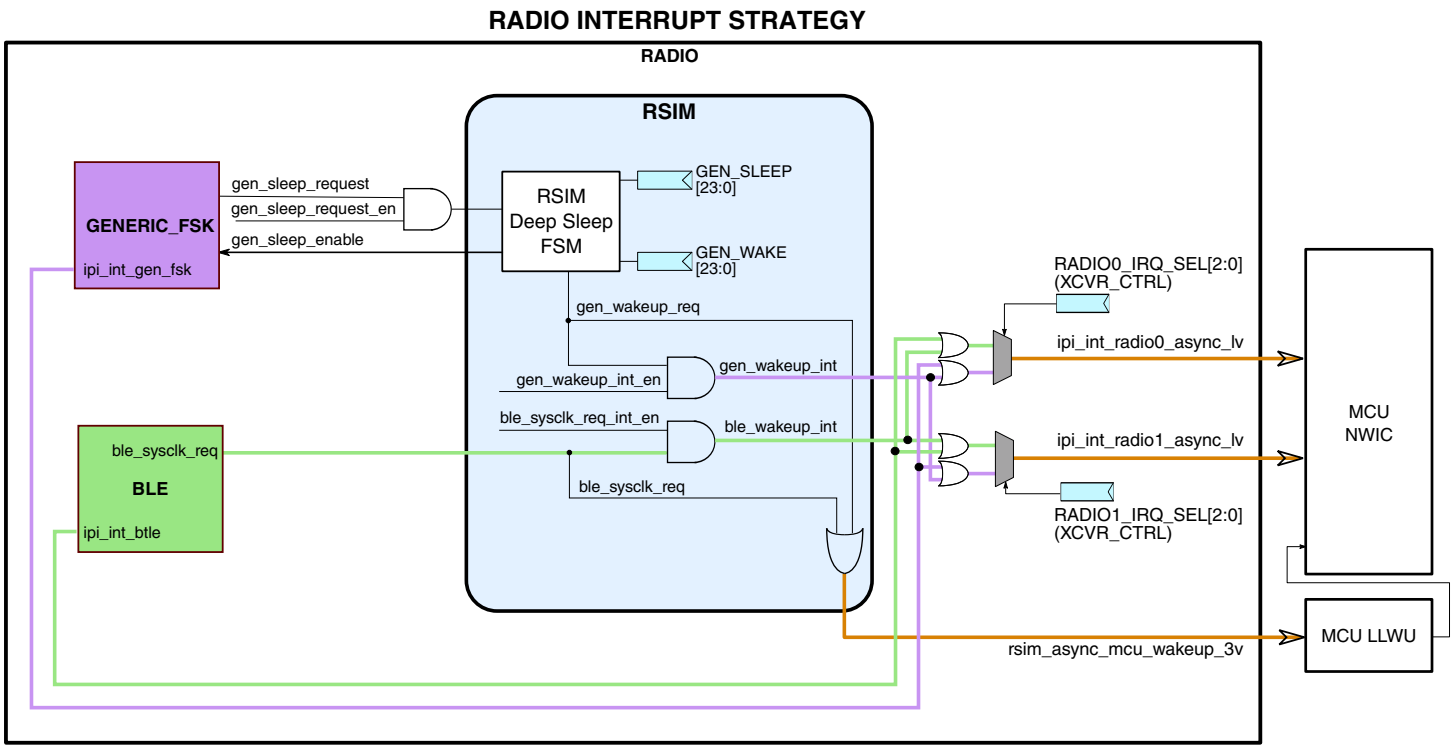


Figure 45-5. Radio Interrupts

45.2.7.2 Enabling Interrupts

There are three places that the interrupt strategy can be programmed:

1. The individual Link Layers interrupts may be configured.
2. The RSIM Deep Sleep Wakeup source may be configured.
3. The XCVR_CTRL register must be configured.

45.2.7.3 XCVR CTRL IRQ Select

Each of the Link Layer interrupts that can be selected here are an OR of the Link Layer interrupt, the WOR wakeup interrupt selected, or the MAN wakeup interrupt selected.

The two OR'd interrupts must be selected using the XCVR_CTRL RADIO0_IRQ_SEL and RADIO1_IRQ_SEL registers in the Transceiver as shown in the following tables:

Assign ipi_int_radio0_async_lv Interrupt (ipi_int_radio0) to a Protocol :

RADIO0_IRQ_SEL	Functionality
0	Assign ipi_int_radio0_async_lv Interrupt to BLE
1	ipi_int_radio0_async_lv Interrupt unassigned
2	ipi_int_radio0_async_lv Interrupt unassigned
3	Assign ipi_int_radio0_async_lv Interrupt to GENERIC_FSK
4-7	ipi_int_radio0_async_lv Interrupt unassigned

Assign ipi_int_radio1_async_lv Interrupt (ipi_int_radio1) to a Protocol :

RADIO1_IRQ_SEL	Functionality
0	Assign ipi_int_radio1_async_lv Interrupt to BLE
1	ipi_int_radio1_async_lv Interrupt unassigned
2	ipi_int_radio1_async_lv Interrupt unassigned
3	Assign ipi_int_radio1_async_lv Interrupt to GENERIC_FSK
4-7	ipi_int_radio0_async_lv Interrupt unassigned

45.2.8 Radio Configuration

The Radio Software Configuration (SW_CONFIG) register contains the software configuration and control bits. These bits can be written by software as part of configuring the Radio and then they can be read as needed to confirm that the Radio has not been reset by either a System Reset or a Power On event.

Please read the Radio Resets section for more information about the reset signals.

The configuration bits are described in the following table:

RSIM Register Field	Function	Description
RADIO_CONFIGURED_POR_RESET	Radio Configuration Bit, cleared by Radio Power On Reset	This is a bit software can write when it wants to be able to check if a Radio Power On Reset occurs.

Table continues on the next page...

RADIO_CONFIGURED_SYS_RESET	Radio Configuration Bit, cleared by Radio System Reset	This is a bit software can write when it wants to be able to check if a Radio System Reset occurs.
BLOCK_SOC_RESETS	Block SoC Resets of the Radio, cleared by Radio System Reset	Most SoC System or Power On Resets will be blocked and the Radio will not be affected by them when this bit is set. The Radio Power Mode logic is not blocked by this bit and will reset on any SoC Power On Reset.
RADIO_RESET_BIT	Software System Reset for the Radio. This bit only clears if there is an SoC Power On Reset.	Setting this bit will reset the Global Radio and RSIM logic, including all Link Layer logic. The logic will stay in reset until this bit is cleared. This bit will not reset the Power Mode logic or the Deep Sleep Mode related logic.
BLOCK_EXT_OSC_PWR_REQ	Block External Requests for RF OSC from starting a Radio Power Wakeup Sequence	If this bit is set then any External request for the OSC will not start a Radio Power Wakeup Sequence, otherwise any such request does request full Run Regulation mode.

45.2.9 Radio Resets

The Radio can be reset by the SoC System Reset, the SoC Power On Reset, or software can block the SoC System Reset and instead perform a Radio Resets using register bits in the RSIM. Please read the Radio Configuration section for descriptions of those bits.

45.2.9.1 Reset Tree Diagram

The Radio Reset Tree is illustrated in the diagram below

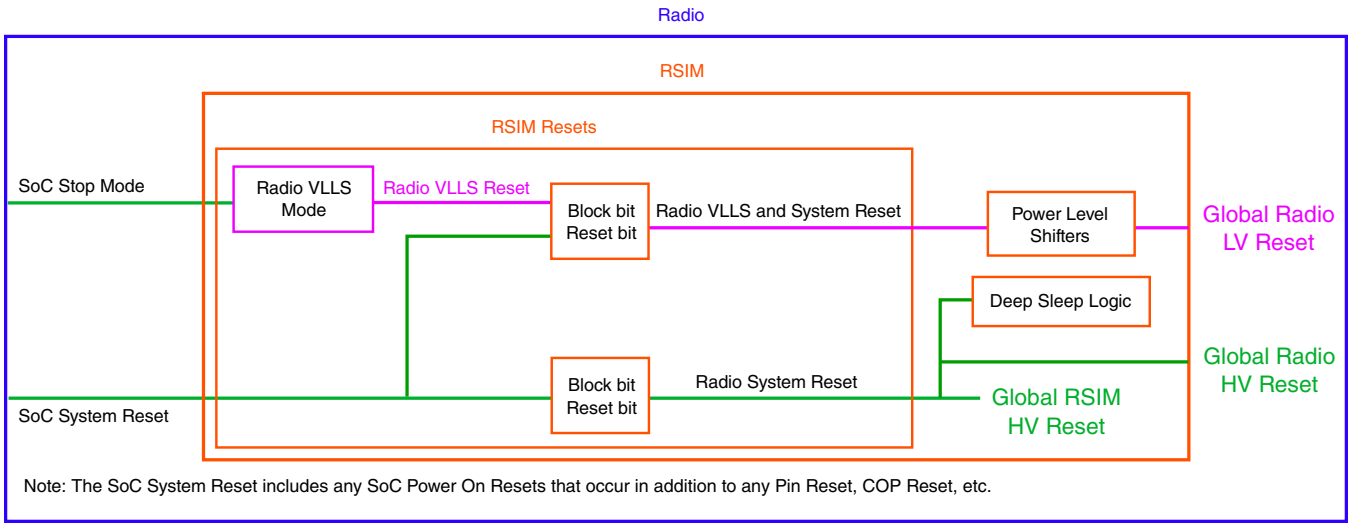


Figure 45-6. Radio Reset Tree

45.2.9.2 Blocking Resets

Software can choose to block SoC resets from reseting the Radio configuration and instead software can then monitor the SoC Resets, and also the Radio Configuration Status using the RADIO_CONFIGURED_SYS_RESET and RADIO_CONFIGURED_POR_RESET bits.

To use this method software should do the following:

- Set RSIM_SW_CONFIG_BLOCK_SOC_RESETS = 1 to start blocking SoC System Resets.
- Set RSIM_SW_CONFIG_RADIO_CONFIGURED_SYS_POR = 1 and monitor it to see if a POR has occurred, which will return all Radio settings to default.
- Monitor the SoC resets and if the Radio needs to be reset use the RSIM_SW_CONFIG_RADIO_RESET_BIT to reset the non-POR only logic.

45.2.10 Radio Gasket Bypass

The SoC platform has an asynchronous gasket that allows register access for the Radio registers in all SoC clocking modes. The RADIO_GASKET_BYPASS_OVRD bit allows software to directly control the SoC platform asynchronous gasket bypass signal.

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 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. 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.

Please also see the Radio Registers Access portion of the Radio Clocks section.

45.2.11 Radio Version ID

The RADIO_VERSION register value can be read to identify the Version of the Radio.

Note: If the Radio digital is in a low power stop mode the Version ID will read as zero.

45.2.12 Memory Map and Register Definition

45.2.12.1 RSIM register descriptions

45.2.12.1.1 RSIM Memory map

RSIM base address: 4005_9000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Radio System Control (CONTROL)	32	RW	00C0_0002h
4h	Deep Sleep Wakeup Sequence (DSM_WAKEUP)	32	RW	0000_0000h
8h	Radio MAC Address (MAC_MSB)	32	RO	0000_0000h
Ch	Radio MAC Address (MAC_LSB)	32	RO	0000_0000h
10h	Radio Miscellaneous (MISC)	32	RW	0800_0000h
18h	Radio Software Configuration (SW_CONFIG)	32	RW	0000_0020h
100h	Deep Sleep Timer (DSM_TIMER)	32	RO	0000_0000h
104h	Deep Sleep Timer Control (DSM_CONTROL)	32	RW	0000_0000h
108h	Deep Sleep Wakeup Time Offset (DSM_OSC_OFFSET)	32	RW	0000_0000h

Table continues on the next page...

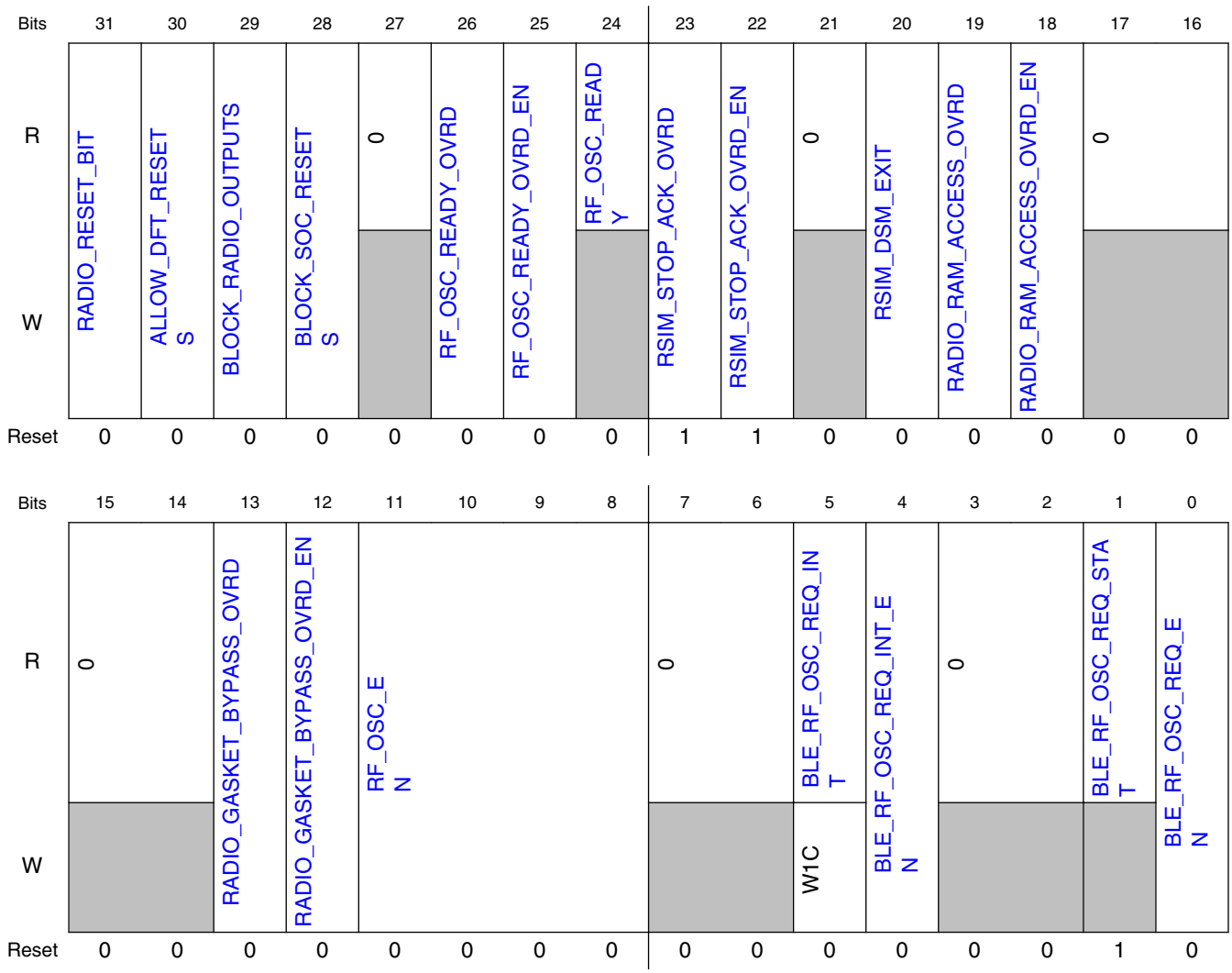
Offset	Register	Width (In bits)	Access	Reset value
11Ch	Generic FSK Link Layer Sleep Time (GEN_SLEEP)	32	RW	0000_0000h
120h	Generic FSK Link Layer Wake Time (GEN_WAKE)	32	RW	0000_0000h
124h	Radio Oscillator Control (RF_OSC_CTRL)	32	RW	A020_3806h
128h	Radio Analog Test Registers (ANA_TEST)	32	RW	0000_0000h
12Ch	Radio Analog Trim Registers (ANA_TRIM)	32	RW	784B_0000h

45.2.12.1.2 Radio System Control (CONTROL)

45.2.12.1.2.1 Offset

Register	Offset
CONTROL	0h

45.2.12.1.2.2 Diagram



45.2.12.1.2.3 Fields

Field	Function
31 RADIO_RESET_BIT	Software Reset for the Radio This bit resets on POR only. When the SoC 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 ALLOW_DFT_RESETS	Allow the DFT Reset Pin to Reset the Radio The Radio provides a port that can be connected to a test mode pin in order to provide a method of resetting the Radio independently from SoC resets. This bit enables that DFT functionality.
29 BLOCK_RADIO_OUTPUTS	Block Radio Outputs

Table continues on the next page...

Field	Function
	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_SOC_RESETS	Block SoC Resets of the Radio 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 SoC resets will be blocked and the Radio will not be affected by them when this bit is set.
27 —	Reserved
26 RF_OSC_READY_OVRD	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.
25 RF_OSC_READY_OVRD_EN	RF Ref Osc Ready Override Enable This bit enables the RF Ref Osc Ready Override bit.
24 RF_OSC_READY	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. The RF Ref Osc Ready signal can be overridden using the RF_OSC_READY_OVRD bit.
23 RSIM_STOP_ACK_OVRD	Stop Acknowledge Override This bit controls the Stop Acknowledge signal to the SoC Core Platform in Override mode.
22 RSIM_STOP_ACK_OVRD_EN	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 Deep Sleep Mode state of the Radio Link Layers and whether the Radio OSC is enabled by any means.
21 —	Reserved
20 RSIM_DSM_EXIT	BLE Force Deep Sleep Mode Exit This bit forces the BLE link layer to wakeup from Deep Sleep Mode.
19 RADIO_RAM_ACCESS_OVRD	Radio RAM Access Override The Radio has two internal RAM blocks that are allowed to go into a low power state when they are not being used by a link layer. If this bit is set, then the RAMs are kept in an active power state to allow software to access them.
18 RADIO_RAM_ACCESS_OVRD_EN	Radio RAM Access Override Enable This bit enables the Radio RAM Access Override bit.
17-16 —	Reserved
15-14 —	Reserved
13	Radio Gasket Bypass Override

Table continues on the next page...

Field	Function
RADIO_GASKET_BYPASS_OVRD	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 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. 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. 0b - XCVR and Link Layer Register Clock is the RF Ref Osc Clock 1b - XCVR and Link Layer Register Clock is the SoC IPG Clock
12 RADIO_GASKET_BYPASS_OVRD_EN	Radio 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.
11-8 RF_OSC_EN	RF Ref Osc Enable Select The RF Reference Oscillator can be enabled by a Radio link layer, by an internal SoC clock mode, by an External Pin request, or by these bits. If these bits are all cleared, 0000, then the RF Ref Osc will be controlled by the SoC, by an external pin request, or by a 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. 0000b - RF Ref Osc will be controlled by the SoC, external pin, or a link layer 0001b - RF Ref Osc on in Run/Wait 0011b - RF Ref Osc on in Stop 0111b - RF Ref Osc on in VLPR/VLPW 1111b - RF Ref Osc on in VLPS
7-6 —	Reserved
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
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.

Field	Function
	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.

45.2.12.1.3 Deep Sleep Wakeup Sequence (DSM_WAKEUP)

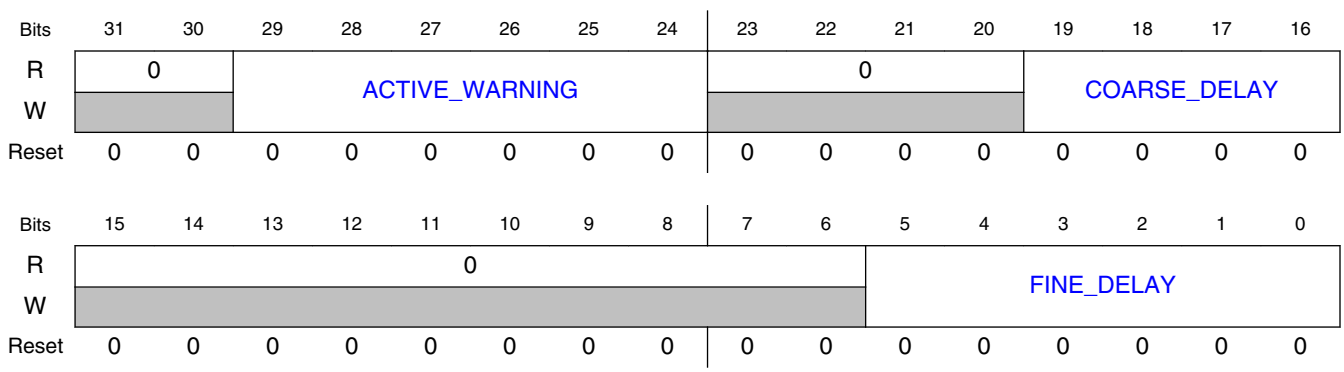
45.2.12.1.3.1 Offset

Register	Offset
DSM_WAKEUP	4h

45.2.12.1.3.2 Function

The RSIM Deep Sleep Wakeup Sequence register provides settings to adjust the RF Osc warm-up timing, and the delay of the RF Active signal used for WiFi coexistence.

45.2.12.1.3.3 Diagram



45.2.12.1.3.4 Fields

Field	Function
31-30 —	Reserved
29-24 ACTIVE_WARNING	Deep Sleep Wakeup RF Active Warning Time This is the delay after RF OSC has been enabled, the range is 0-63 and the resulting delay is 122.07 us x ACTIVE_WARNING + 30.5 us. This is typically when the RF Osc should be stable.
23-20 —	Reserved

Table continues on the next page...

Field	Function
19-16 COARSE_DELAY	Deep Sleep Wakeup Coarse Delay Time This is the first delay after the beginning of the Wake-Up sequence, the range is 0-15 and the resulting delay is 1.953 ms x COARSE_DELAY + 61 us, with a minimum delay of 61 us.
15-6 —	Reserved
5-0 FINE_DELAY	Deep Sleep Wakeup Fine Delay Time This is the second delay after the beginning of the Wake-Up sequence, the range is 0-63 and the resulting delay is 122.07 us x FINE_DELAY + 61 us, with a minimum delay of 61 us. This is when the RF OSC is enabled.

45.2.12.1.4 Radio MAC Address (MAC_MSB)

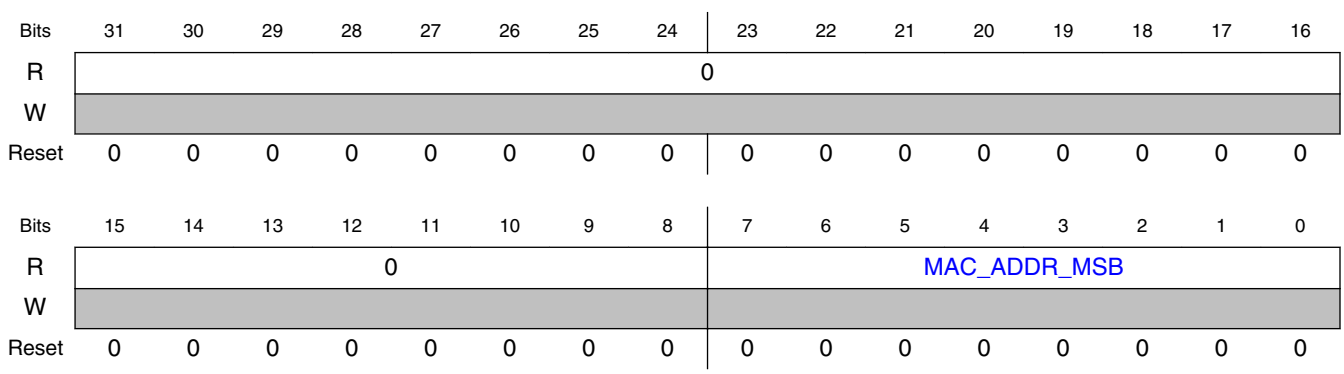
45.2.12.1.4.1 Offset

Register	Offset
MAC_MSB	8h

45.2.12.1.4.2 Function

The Radio MAC Address registers provide a unique ID that is stored in the Flash during factory test.

45.2.12.1.4.3 Diagram



45.2.12.1.4.4 Fields

Field	Function
31-8	Reserved

Table continues on the next page...

Field	Function
—	
7-0 MAC_ADDR_MSB	Radio MAC Address MSB The Radio 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.

45.2.12.1.5 Radio MAC Address (MAC_LSB)

45.2.12.1.5.1 Offset

Register	Offset
MAC_LSB	Ch

45.2.12.1.5.2 Function

The Radio MAC Address registers provide a unique ID that is stored in the Flash during factory test

45.2.12.1.5.3 Diagram

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

45.2.12.1.5.4 Fields

Field	Function
31-0 MAC_ADDR_LSB	Radio MAC Address LSB The Radio 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.

45.2.12.1.6 Radio Miscellaneous (MISC)

45.2.12.1.6.1 Offset

Register	Offset
MISC	10h

45.2.12.1.6.2 Diagram

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

45.2.12.1.6.3 Fields

Field	Function
31-24 RADIO_VERSION	Radio Version ID number This register value can be read to identify the Version of the Radio. Note that the value will read as zero unless the Radio is in Run Mode. B = 2.4 GHz Radio 2.0 8 = 2.4 GHz Radio 2.1 4 = 2.4 GHz Radio 3.0 5 = 2.4 GHz Radio 3.1
23-20 —	Reserved
19-16 —	Reserved
15-5 —	Reserved
4-0 —	Reserved

45.2.12.1.7 Radio Software Configuration (SW_CONFIG)

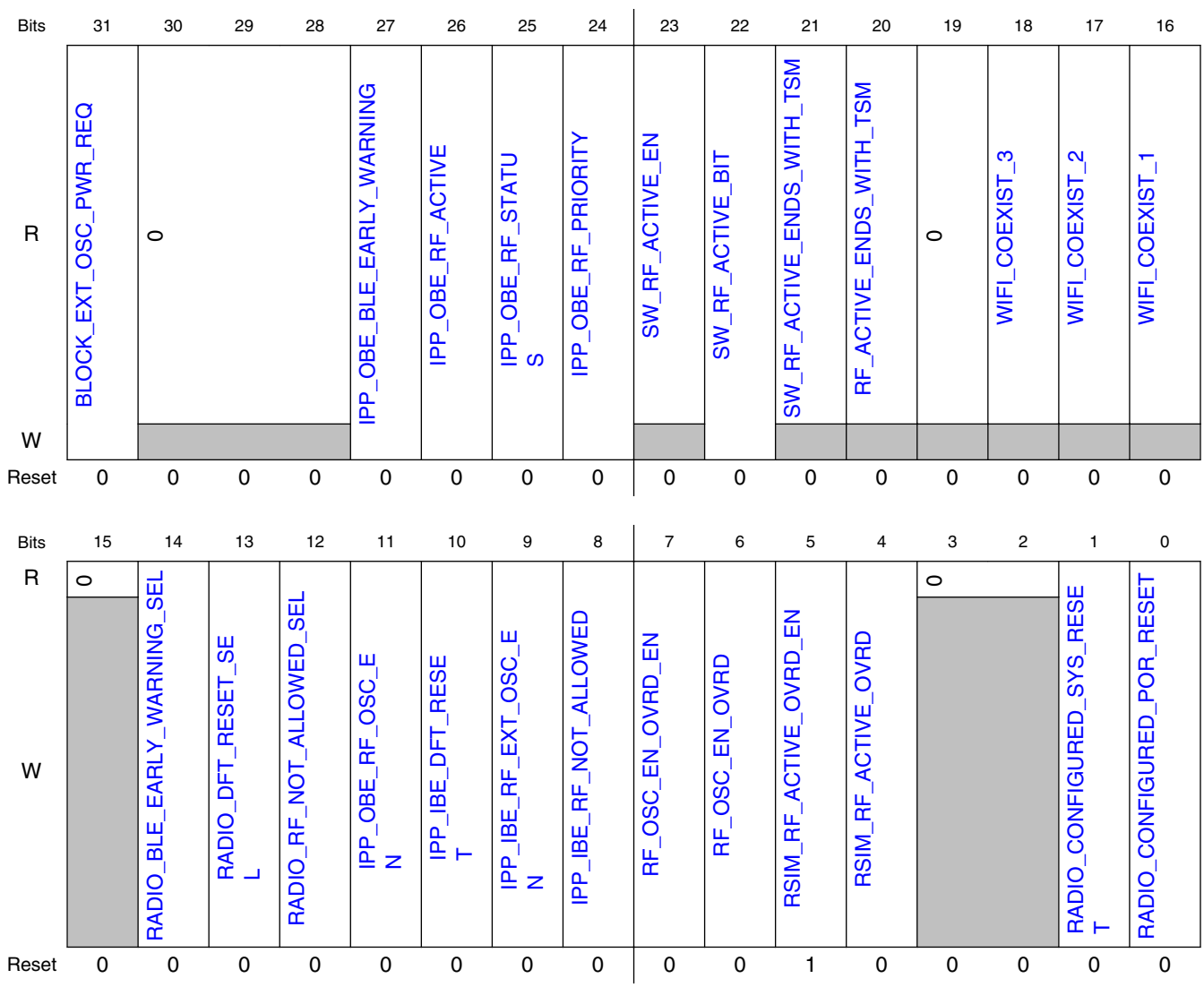
45.2.12.1.7.1 Offset

Register	Offset
SW_CONFIG	18h

45.2.12.1.7.2 Function

This register contains the software configuration and control bits.

45.2.12.1.7.3 Diagram



45.2.12.1.7.4 Fields

Field	Function
31 BLOCK_EXT_OSC_PWR_REQ	Block External Requests for RF OSC from starting a Radio Power Wakeup Sequence If this bit is set then any External request for the OSC will not start a Radio Power Wakeup Sequence, otherwise any such request does request full Run Regulation mode.
30-28 —	Reserved
27 IPP_OBE_BLE_EARLY_WARNING	IPP_OBE_BLE_EARLY_WARNING This bit enables the Output Driver (OBE) on the SoC port that provides the BLE Early Warning signal as a pin interface option.
26 IPP_OBE_RF_ACTIVE	IPP_OBE_RF_ACTIVE This bit enables the Output Driver (OBE) on the SoC port that provides the RF Active signal as a pin interface option.
25 IPP_OBE_RF_STATUS	IPP_OBE_RF_STATUS This bit enables the Output Driver (OBE) on the SoC port that provides the RF Status signal as a pin interface option.
24 IPP_OBE_RF_PRIORITY	IPP_OBE_RF_PRIORITY This bit enables the Output Driver (OBE) on the SoC port that provides the RF Priority signal as a pin interface option.
23 SW_RF_ACTIVE_EN	Software RF_ACTIVE Control Enable If this bit is set the SW_RF_ACTIVE_BIT acts as an OR source for the RF_ACTIVE pin.
22 SW_RF_ACTIVE_BIT	Software RF_ACTIVE Control Bit If the SW_RF_ACTIVE_EN bit is set this bit acts as an OR source for the RF_ACTIVE pin.
21 SW_RF_ACTIVE_ENDS_WITH_TSM	Software RF_ACTIVE clearing mechanism If this bit is set along with the SW_RF_ACTIVE_EN bit, then the Software RF_ACTIVE OR source will clear when the TSM sequence ends.
20 RF_ACTIVE_ENDS_WITH_TSM	RF_ACTIVE clearing mechanism If this bit is set then the RSIM RF_ACTIVE OR source will clear when the TSM sequence ends.
19 —	Reserved
18 WIFI_COEXIST_3	RF_EARLY_WARNING Source If this bit is set the RF_EARLY_WARNING pin will be controlled by TSM Spare 3, otherwise it will be controlled by the BLE Link Layer.
17 WIFI_COEXIST_2	RF_STATUS Source If this bit is set the RF_STATUS pin will be controlled by TSM Spare 3, otherwise it will be controlled by TSM Spare 2.
16 WIFI_COEXIST_1	RF_ACTIVE Source If this bit is set the RF_ACTIVE pin will be controlled by TSM Spare 1, otherwise it will be controlled by the RSIM RF Active Sequence along with any override by the SW_RF_ACTIVE_BIT.

Table continues on the next page...

Field	Function
15 —	Reserved
14 RADIO_BLE_EARLY_WARNING_SEL	Radio BLE_EARLY_WARNING Select The Radio has two pins which may be used as an early warning of a BLE activity. This bit selects which of the two pins has that early warning.
13 RADIO_DFT_RESET_SEL	Radio DFT_RESET Select The Radio has two pins which may be used as a DFT_RESET. This bit selects the two pins.
12 RADIO_RF_NOT_ALLOWED_SEL	Radio RF_NOT_ALLOWED Select The Radio has two pins which may be controlled by external pins in the SoC in such a way as to allow an external device, such as another radio, to coexist with this Radio. This bit selects which of the two pins has control of that external request.
11 IPP_OBE_RF_OSC_EN	IPP_OBE_RF_OSC_EN This bit enables the Output Driver (OBE) on the SoC port that provides the RF Osc Enable signal as a pin interface option.
10 IPP_IBE_DFT_RESET	IPP_IBE_DFT_RESET This bit enables the Input Driver (IBE) on the SoC port that provides the DFT_RESET as a pin interface option.
9 IPP_IBE_RF_EXT_OSC_EN	IPP_IBE_RF_EXT_OSC_EN This bit enables the Input Driver (IBE) on the SoC port that provides the RF_EXT_OSC_EN as a pin interface option.
8 IPP_IBE_RF_NOT_ALLOWED	IPP_IBE_RF_NOT_ALLOWED This bit enables the Input Driver (IBE) on the SoC port that provides the RF_NOT_ALLOWED as a pin interface option.
7 RF_OSC_EN_OVRD_EN	Radio Osc Enable Override Enable This bit enables the Radio Osc Enable Override bit.
6 RF_OSC_EN_OVRD	Radio Osc Enable Override This bit directly controls the RF_OSC_EN SoC pin signal when the Radio Osc Enable Override is enabled.
5 RSIM_RF_ACTIVE_OVRD_EN	RF Active Internal Override Enable This bit enables the RF Active Internal Override bit.
4 RSIM_RF_ACTIVE_OVRD	RF Active Internal Override If the RSIM_RF_ACTIVE_OVRD_EN bit is set, then this bit controls the Internal version of RF Active instead of the Link Layers. The Internal version of RF Active whether overridden or not has a lower priority and can always be overridden by software using the SW_RF_ACTIVE_BIT or the TSM Spare 1 bit if it is selected using the WIFI_COEXIST_1 select bit.
3-2 —	Reserved
1 RADIO_CONFIGURED_SYS_RESET	Radio Configuration Bit, cleared by Radio System Reset This is a bit software can write when it wants to be able to check if a Radio System Reset occurs.

Table continues on the next page...

Field	Function
0 RADIO_CONFIGURED_POR_RESET	Radio Configuration Bit, cleared by Radio Power On Reset This is a bit software can write when it wants to be able to check if a Radio Power On Reset occurs.

45.2.12.1.8 Deep Sleep Timer (DSM_TIMER)

45.2.12.1.8.1 Offset

Register	Offset
DSM_TIMER	100h

45.2.12.1.8.2 Diagram

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

45.2.12.1.8.3 Fields

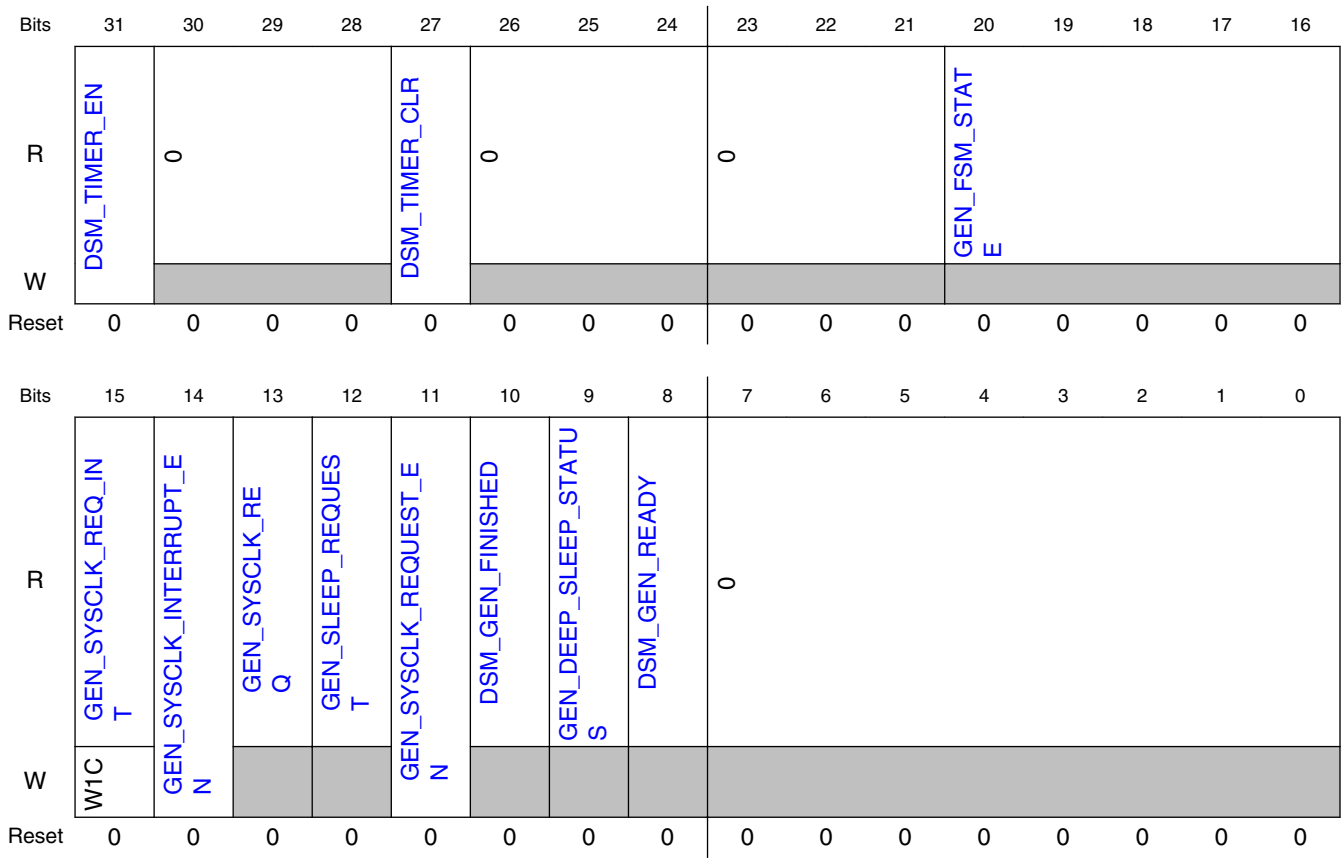
Field	Function
31-24 —	Reserved
23-0 DSM_TIMER	Deep Sleep Mode Timer This read only register shows the value of the Deep Sleep Timer. The timer counts clock cycles of the 32 kHz SoC oscillator whenever the DSM_TIMER_EN bit is set. The timer is reset to zero whenever the DSM_TIMER_CLR bit is set.

45.2.12.1.9 Deep Sleep Timer Control (DSM_CONTROL)

45.2.12.1.9.1 Offset

Register	Offset
DSM_CONTROL	104h

45.2.12.1.9.2 Diagram



45.2.12.1.9.3 Fields

Field	Function
31 DSM_TIMER_EN	Deep Sleep Mode Timer Enable Whenever this bit is set the Deep Sleep Mode Timer counts clock cycles of the 32 kHz SoC oscillator.
30-28 —	Reserved
27 DSM_TIMER_CLR	Deep Sleep Mode Timer Clear Whenever this bit is set the Deep Sleep Mode Timer is reset to zero.

Table continues on the next page...

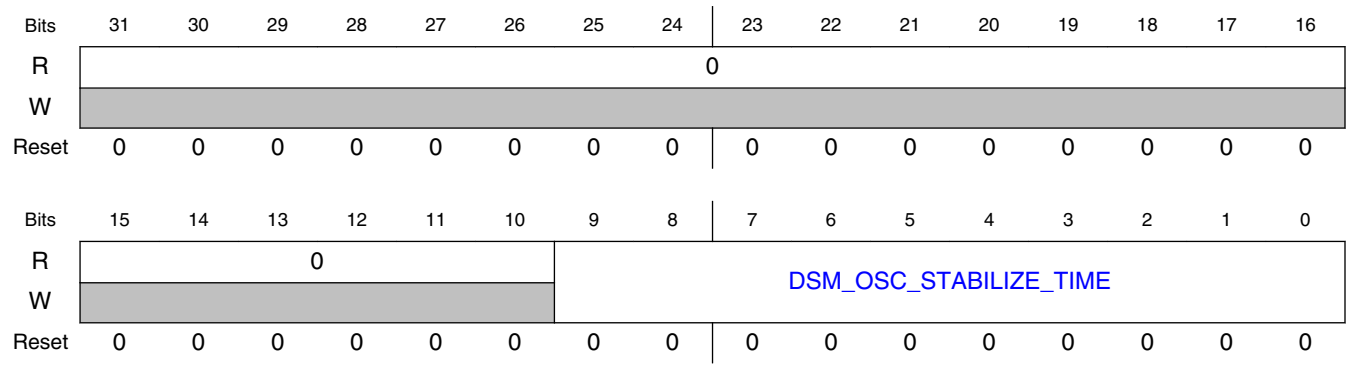
Field	Function
26-24 —	Reserved
23-21 —	Reserved
20-16 GEN_FSM_STATE	GEN Deep Sleep State Machine State This is the current State that the GEN Deep Sleep Finite State Machine is in.
15 GEN_SYSCLK_REQ_INT	Interrupt Flag from an Generic FSK Link Layer RF OSC Request If the GEN_SYSCLK_INTERRUPT_EN bit is set, then this flag indicates that an Generic FSK RF OSC Request has occurred. The flag will persist until this bit is written with a 1 to clear it.
14 GEN_SYSCLK_INTERRUPT_EN	Generic FSK Link Layer RF OSC Request Interrupt Enable If this bit is set, then a change from a 0 to a 1 on GEN_SYSCLK_REQ will generate an interrupt which can be used by the SoC.
13 GEN_SYSCLK_REQ	Generic FSK Link Layer RF OSC Request Status This bit shows the status of the Generic FSK Link Layer Request for the RF OSC to be turned on. If the GEN_SYSCLK_REQUEST_EN bit is set, then this signal is used by the RSIM to turn on the RF OSC if the RF OSC is not already turned on by another request source.
12 GEN_SLEEP_REQUEST	Generic FSK Link Layer Deep Sleep Requested From the Generic FSK Sleep Enable register. This enables a match of GEN_SLEEP[23:0] to SLEEP_TMR[23:0]; when the match occurs Deep Sleep Mode is entered.
11 GEN_SYSCLK_REQUEST_EN	Enable Generic FSK Link Layer to Request RF OSC This bit allows the Generic FSK Link Layer to request turning on the RF OSC.
10 DSM_GEN_FINISHED	Generic FSK Deep Sleep Time Finished This is the SLEEP_FINISHED state in the Deep Sleep Module State Machine, the state machine holds here until the Link Layer de-asserts the sleep_request.
9 GEN_DEEP_SLEEP_STATUS	Generic FSK Link Layer Deep Sleep Mode Status This is the signal from the Deep Sleep Module State Machine telling the Link Layer to enter and exit Deep Sleep Mode. If this bit is set then the Link Layer is in Deep Sleep Mode.
8 DSM_GEN_READY	Generic FSK Ready for Deep Sleep Mode This is the SLEEP_READY state in the Deep Sleep Module State Machine, the state machine holds here until the sleep time is reached on the 32 kHz clock counter (Deep Sleep Timer).
7-0 —	Reserved

45.2.12.1.10 Deep Sleep Wakeup Time Offset (DSM_OSC_OFFSET)

45.2.12.1.10.1 Offset

Register	Offset
DSM_OSC_OFFSET	108h

45.2.12.1.10.2 Diagram



45.2.12.1.10.3 Fields

Field	Function
31-10 —	Reserved
9-0 DSM_OSC_STABILIZE_TIME	Deep Sleep Wakeup RF OSC Stabilize Time This register should be programmed by software with the time needed for the RF OSC to stabilize to its specified accuracy after it is enabled. This time is represented by the number of clock cycles of the 32 kHz SoC oscillator that should be counted before the link layer is allowed to exit Deep Sleep Mode on a wakeup event. This time offset will be used by the Deep Sleep State Machine to first turn on the RF OSC, then wait for it stabilize, and finally wakeup the link layer.

45.2.12.1.11 Generic FSK Link Layer Sleep Time (GEN_SLEEP)

45.2.12.1.11.1 Offset

Register	Offset
GEN_SLEEP	11Ch

45.2.12.1.11.2 Diagram

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

45.2.12.1.11.3 Fields

Field	Function
31-24 —	Reserved
23-0 GEN_SLEEP_TIME	Generic FSK Link Layer Sleep Time Software determines the future time at which it would like to enter Deep Sleep Mode by reading the DSM_TIMER, computing the number of 32 kHz clock cycles remaining until the desired DSM start-time, and writing that value into this register. The value programmed into this register should be no fewer than 4 clocks greater (in the future) than the current time as read from DSM_TIMER.

45.2.12.1.12 Generic FSK Link Layer Wake Time (GEN_WAKE)

45.2.12.1.12.1 Offset

Register	Offset
GEN_WAKE	120h

45.2.12.1.12.2 Diagram

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

45.2.12.1.12.3 Fields

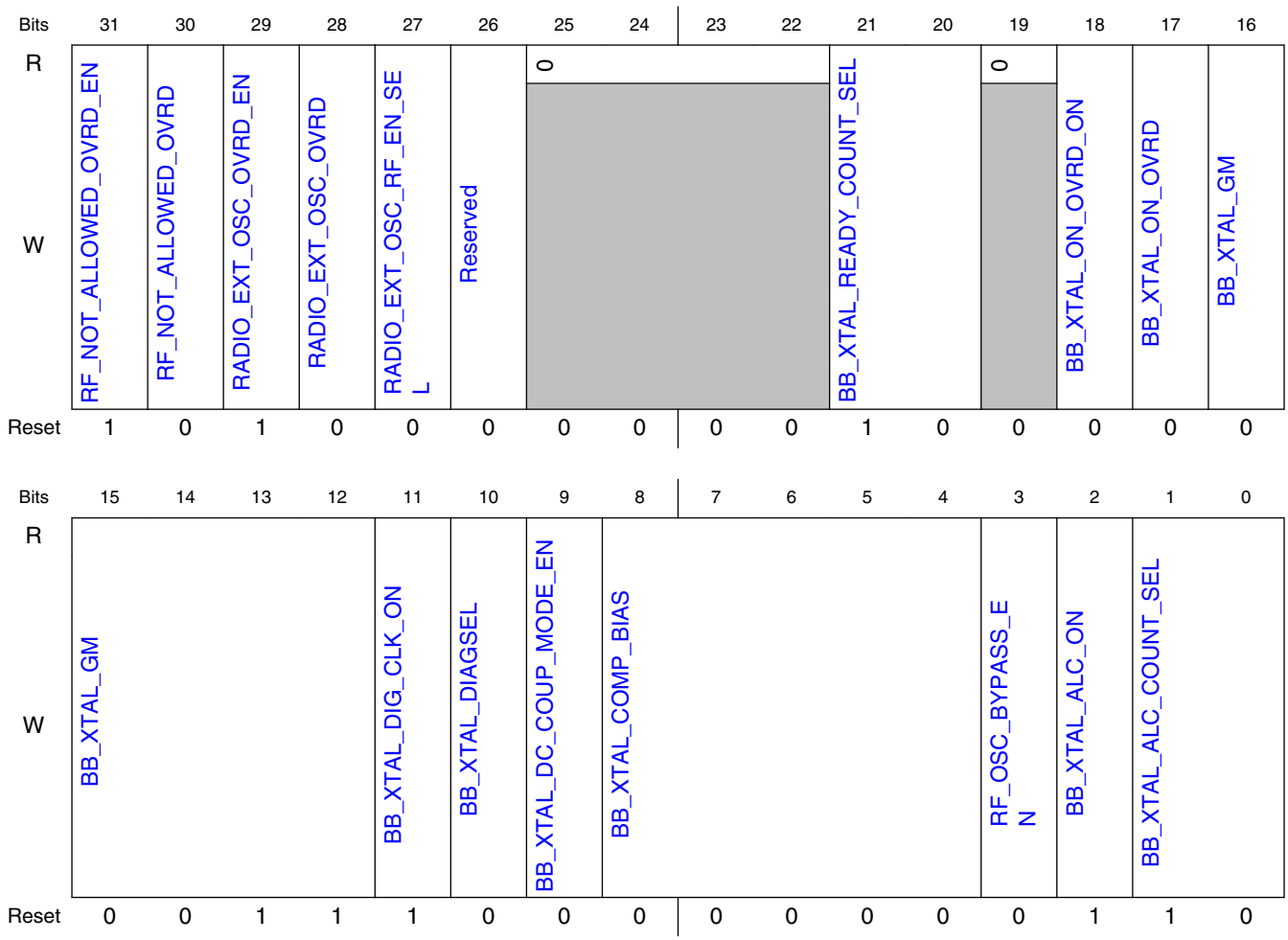
Field	Function
31-24 —	Reserved
23-0 GEN_WAKE_TIME	Generic FSK Link Layer Wake Time Software determines the future time at which it would like to exit Deep Sleep Mode by reading the DSM_TIMER, computing the number of 32 kHz clock cycles remaining until the desired DSM exit time, and writing that value into this register.

45.2.12.1.13 Radio Oscillator Control (RF_OSC_CTRL)

45.2.12.1.13.1 Offset

Register	Offset
RF_OSC_CTRL	124h

45.2.12.1.13.2 Diagram



45.2.12.1.13.3 Fields

Field	Function
31 RF_NOT_ALLOWED_OVRD_EN	RF Not Allowed Override Enable This bit enables the RF Not Allowed Override bit.
30 RF_NOT_ALLOWED_OVRD	RF Not Allowed Override If the RF_NOT_ALLOWED_OVRD_EN bit is set, then this bit controls the RF Not Allowed functionality instead of the SoC pin.
29 RADIO_EXT_OSC_OVRD_EN	Radio External Request for RF OSC Override Enable This bit enables the Radio External RF OSC Override bit.
28 RADIO_EXT_OSC_OVRD	Radio External Request for RF OSC Override If the RADIO_EXT_OSC_OVRD_EN bit is set, then this bit controls that External RF OSC Request functionality.

Table continues on the next page...

Field	Function
27 RADIO_EXT_OSC_RF_EN_SEL	Radio External Request for RF OSC Select The Radio has two pins which may be controlled by external pins in the SoC in such a way as to allow an external device, such as another radio, to enable this Radio's RF Oscillator. This bit selects which of the two pins has control of that external request.
26 —	Reserved
25-22 —	Reserved
21-20 BB_XTAL_READY_COUNT_SEL	rmap_bb_xtal_ready_count_sel_hv[1:0] Program counter for xtal ready signal Sets up count value for XO startup time. 00b - 1024 counts (32 us @ 32 MHz) 01b - 2048 (64 us @ 32 MHz) 10b - 4096 (128 us @ 32 MHz) 11b - 8192 (256 us @ 32 MHz)
19 —	Reserved
18 BB_XTAL_ON_OVRD_ON	rmap_bb_xtal_on_ovrd_on_hv Enable override XO enable bit Enable selector: 0b - rfctrl_bb_xtal_on_hv is asserted 1b - rfctrl_bb_xtal_on_ovrd_hv is asserted
17 BB_XTAL_ON_OVRD	rmap_bb_xtal_on_ovrd_hv Override XO enable
16-12 BB_XTAL_GM	rmap_bb_xtal_gm_hv[4:0] Amplifier current bumps, bit [4] not used. Current values assume ALC is off 0 - Min setting: 8 current sources on (TT27:60uA) ... 15 - Max setting: 128 current sources on (TT27:960uA)
11 BB_XTAL_DIG_CLK_ON	rmap_bb_xtal_dig_clk_on_hv Enable digital clk output
10 BB_XTAL_DIAGSEL	rmap_bb_xtal_diagsel_hv Enable diagnostics for XO block
9 BB_XTAL_DC_COUP_MODE_EN	rmap_bb_xtal_dc_coup_mode_en_hv This bit enables the external dc coupled mode. This bit powers down the XO amplifier to enable DC coupled input.
8-4 BB_XTAL_COMP_BIAS	rmap_bb_xtal_comp_bias_hv[4:0] Not used.

Table continues on the next page...

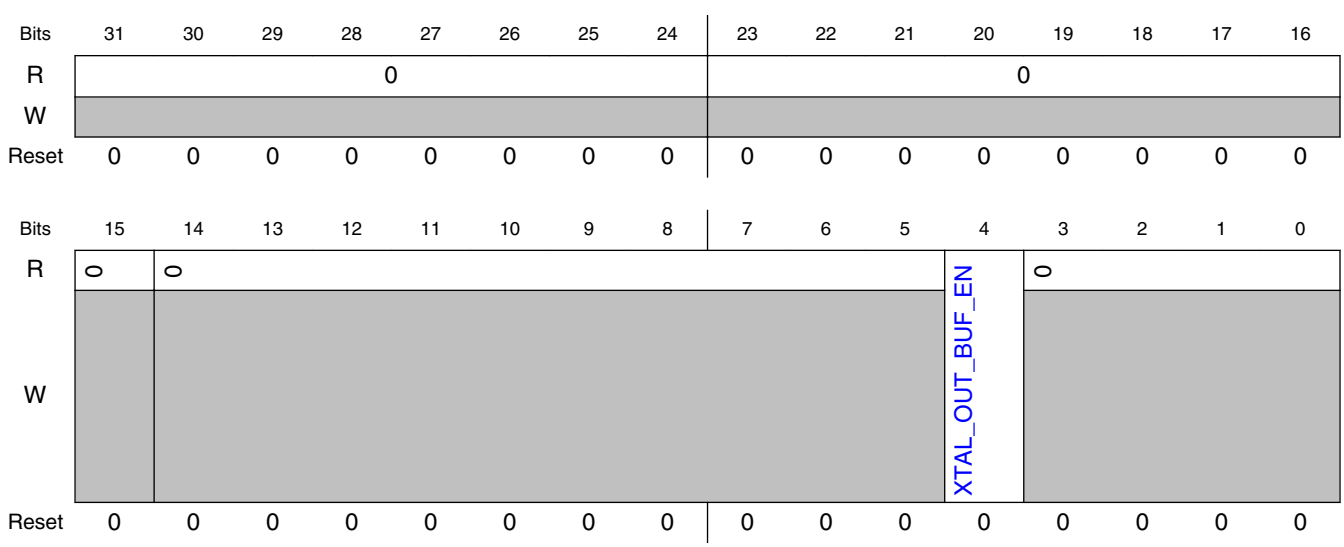
Field	Function
3 RF_OSC_BYPASS_EN	RF Ref Osc Bypass Enable This bit engages the RF Ref Osc analog bypass circuit if the RF Ref Osc is enabled. When the RF Ref Osc is in bypass mode it passes the RF EXTAL clock as the RF Ref Osc clock. Note that the RF Ref Osc Ready signal functions normally in RF OSC Bypass mode, unless overridden with the RF_OSC_READY_OVRD_EN bit.
2 BB_XTAL_ALC_ON	rmap_bb_xtal_alc_on_hv Enable ALC
1-0 BB_XTAL_ALC_COUNT_SEL	rmap_bb_xtal_alc_count_sel_hv[1:0] Program counter for alc ready signal Sets up count value for fastcharge to turn off: 00b - 2048 (64 us @ 32 MHz) 01b - 4096 (128 us @ 32 MHz) 10b - 8192 (256 us @ 32 MHz) 11b - 16384 (512 us @ 32 MHz)

45.2.12.1.14 Radio Analog Test Registers (ANA_TEST)

45.2.12.1.14.1 Offset

Register	Offset
ANA_TEST	128h

45.2.12.1.14.2 Diagram



45.2.12.1.14.3 Fields

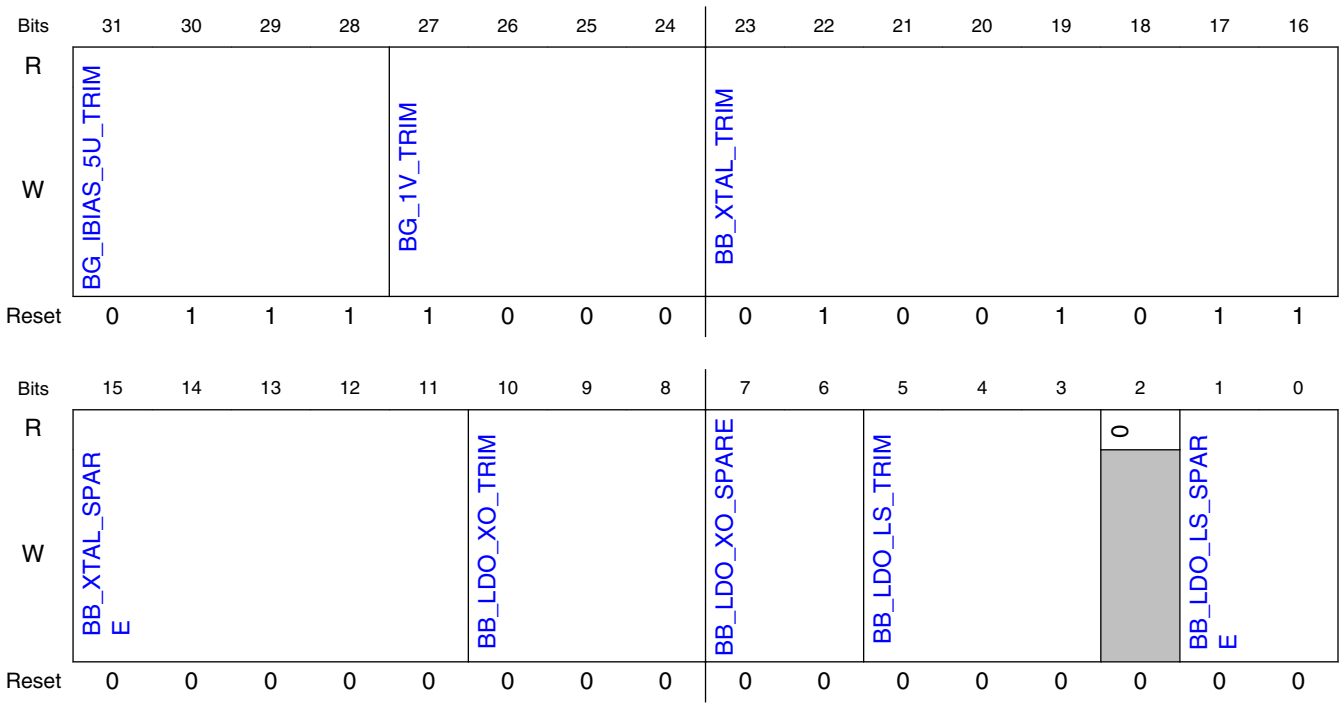
Field	Function
31-24 —	Reserved
23-15 —	Reserved
14-5 —	Reserved.
4 XTAL_OUT_BUF_EN	XTAL Output Buffer Enable The RF Analog Oscillator XTAL_OUT buffer for the XTAL_OUT SoC Pin is enabled when this bit is set. Note that this bit only enables the buffer to the external pin, it does not enable XTAL operation. That must be done through either an internal SOC request or external request via RF_RFOSC_EN pin.
3-0 —	Reserved.

45.2.12.1.15 Radio Analog Trim Registers (ANA_TRIM)

45.2.12.1.15.1 Offset

Register	Offset
ANA_TRIM	12Ch

45.2.12.1.15.2 Diagram



45.2.12.1.15.3 Fields

Field	Function
31-28 BG_IBIAS_5U_TRIM	rmap_bg_ibias_5u_trim_hv[3:0] 5uA current trim bits. Default setting is 0111 Trim bits for VBG output current 0000b - 3.55 uA 0001b - 3.73 uA 0010b - 4.04 uA 0011b - 4.22 uA 0100b - 4.39 uA 0101b - 4.57 uA 0110b - 4.89 uA 0111b - 5.06 (Default) 1000b - 5.23 uA 1001b - 5.41 uA 1010b - 5.72 uA 1011b - 5.9 uA 1100b - 6.07 uA 1101b - 6.25 uA 1110b - 6.56 uA 1111b - 6.74 uA
27-24 BG_1V_TRIM	rmap_bg_1v_trim_hv[3:0] Trim bits for VBG output voltage Trim bits for VBG output voltage

Table continues on the next page...

Field	Function
	0000b - 954.14 mV 0001b - 959.26 mV 0010b - 964.38 mV 0011b - 969.5 mV 0100b - 974.6 mV 0101b - 979.7 mV 0110b - 984.8 mV 0111b - 989.9 mV 1000b - 995 mV (Default) 1001b - 1 V 1010b - 1.005 V 1011b - 1.01 V 1100b - 1.015 V 1101b - 1.02 V 1110b - 1.025 V 1111b - 1.031 V
23-16 BB_XTAL_TRIM	rmap_bb_xtal_trim_hv[7:0] Bump XO load capacitor Load capacitor bumps. bit [7] not used. 0 - Min C1: 5.7pF. Min C2: 7.1pF ... 127 - Max C1: 22.6pF. Max C2: 28.2pF
15-11 BB_XTAL_SPARE	rmap_bb_xtal_spare_hv[4:0] Bit 4: Override RF Analog XTAL Ready signal ; 0 - No override, BB_XTAL_READY_COUNT_SEL is used ; 1- Force RF Analog XTAL Ready = 1. Note that this RF Analog signal can be also be overridden using the RSIM RF_OSC_READY_OVRD bits. Bit 3: XTAL_OUT output Polarity invert bit; 0 - No Polarity inversion; 1 - Polarity inverted Bit 2: This bit selects the clock to be output on the XTAL_OUT SoC pin. The default is to select the XTAL as the clock source. If this bit is written to a 1 then the AuxPLL output will be selected instead. Bits 1:0 are unused.
10-8 BB_LDO_XO_TRIM	rmap_bb_ldo_xo_trim_hv[2:0] Trim settings for LDO. Trim settings for LDO. 000b - 1.20 V (Default) 001b - 1.25 V 010b - 1.28 V 011b - 1.33 V 100b - 1.40 V 101b - 1.44 V 110b - 1.50 V 111b - 1.66 V
7-6 BB_LDO_XO_SPARE	rmap_bb_ldo_xo_spare_hv[1:0] Spare bits for LDO, not used.
5-3 BB_LDO_LS_TRIM	rmap_bb_ldo_ls_trim_hv[2:0] Trim settings for LDO. Trim settings for LDO.

Table continues on the next page...

Field	Function
	000b - 1.20 V (Default) 001b - 1.25 V 010b - 1.28 V 011b - 1.33 V 100b - 1.40 V 101b - 1.44 V 110b - 1.50 V 111b - 1.66 V
2 —	Reserved
1-0 BB_LDO_LS_SPARE	rmap_bb_ldo_ls_spare_hv[1:0] Spare bits. Not used so far.

45.3 Transceiver Digital

45.3.1 Transmitter Digital Module

45.3.1.1 About the Transmitter Digital

The Transmitter Digital processes the Transmission Data from the RF Protocol Link Layers and presents the processed data to the PLL Frequency Synthesizer as the Baseband Frequency Word.

45.3.1.1.1 Block diagram

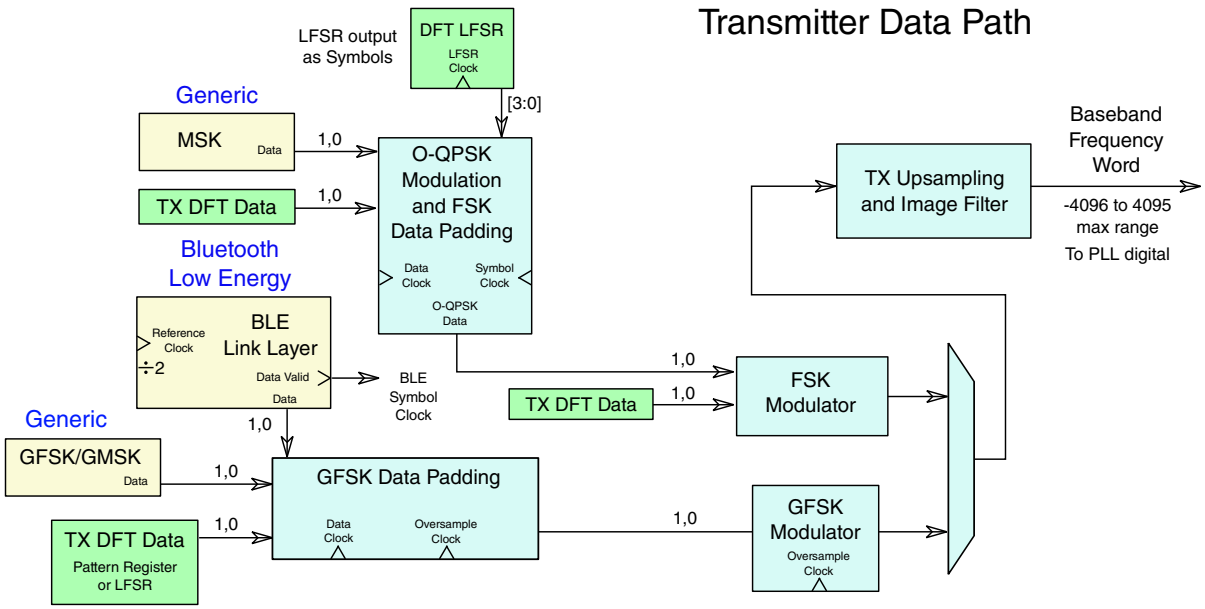


Figure 45-7. Transmitter Digital

45.3.1.1.2 Modulation Types

Modulation Type[1]	Data/Chip Rate (kbps)[2]	BT Product[3]	Modulation Index[4]	Symbol Time (us)	Max Frequency Deviation (kHz)
GMSK/ GFSK,BT=0.5, h=0.5	2000	0.5	0.5	0.5	500
	1000	0.5	0.5	1	250
	500	0.5	0.5	2	125
	250	0.5	0.5	4	62.5
GFSK, BT=0.5, h=0.32	2000	0.5	0.32	0.5	320
	1000	0.5	0.32	1	160
	500	0.5	0.32	2	80
	250	0.5	0.32	4	40
GFSK, BT=0.5, h=0.7	1000	0.5	0.7	1	350
	500	0.5	0.7	2	175
	250	0.5	0.7	4	87.5
GFSK, BT=0.5, h=1.0	1000	0.5	1	1	500
	500	0.5	1	2	250
	250	0.5	1	4	125
GMSK, BT=0.3	2000	0.3	0.5	0.5	300
	1000	0.3	0.5	1	150
	500	0.3	0.5	2	75
	250	0.3	0.5	4	37.5

Table continues on the next page...

GMSK, BT=0.7	2000	0.7	0.5	0.5	700
	1000	0.7	0.5	1	350
	500	0.7	0.5	2	175
	250	0.7	0.5	4	87.5
Generic MSK	2000	NA	0.5	0.5	500
	1000	NA	0.5	1	250
	500	NA	0.5	2	125
	250	NA	0.5	4	62.5
Generic FSK	2000	NA	0.5	0.5	500
	1000	NA	0.5	1	250
	500	NA	0.5	2	125
	250	NA	0.5	4	62.5

1. Modulation Type is set using the PROTOCOL bits in the XCVR_CTRL register.
2. Data Rate is set using the BITRATE bits in the GENFSK register.
3. Bandwidth Time Product is changed by reprogramming the GFSK Gaussian Filter coefficients.
4. Modulation Index is set using the GFSK_MI register bits.

45.3.1.1.3 Data Rates

For BLE protocols the TX Data Rate is set automatically.

For other Radio protocols, the Transit Data Rate is set by the BITRATE bits in the GENFSK register.

The supported data rates are shown in the table below.

Table 45-4. Transmitter Data Rates

GENFSK BITRATE[1:0] Register	TX Data Rate
0	1000 kbit/sec
1	500 kbit/sec
2	250 kbit/sec
3	2000 kbit/sec

45.3.1.1.4 Over-Sampling

The TX Digital uses an Oversample Clock to process the transmission data.

The GFSK, GMSK, and MSK Oversample Ratios are summarized below.

Table 45-5. Oversample Ratios for GFSK, GMSK, and MSK Modulations

TX Data Rate (kb/s)	Ref Clk Freq (MHz)	OSR Ref Clk Divider	OSR Clk Rate (MHz)	Data Oversample Rate

Table continues on the next page...

Table 45-5. Oversample Ratios for GFSK, GMSK, and MSK Modulations (continued)

2000	32	2	16	8
1000	32	4	8	8
500	32	8	4	8
250	32	16	2	8
2000	26	1	26	13
1000	26	2	13	13
500	26	4	6.5	13
250	26	8	3.25	13

45.3.1.1.5 Frequency Word Adjust

The Transmitter has a frequency adjustment register, FREQ_WORD_ADJ, which is used as a signed 9-bit number that is added to the TX Digital output before it is presented to the PLL as the baseband frequency word. This allows the baseband frequency word to be adjusted, or skewed, by a range of -512 to +511.

This frequency adjustment is applied to the final modulation word from any source (GFSK, FSK, DFT) and there is no protection from a math overflow. So the baseband frequency word range of -4096 to 4095 must be considered when adding this frequency word adjustment.

45.3.1.2 Data Pre-Processing

Before the Transmission Data is processed by the GFSK or FSK Modulators, it can be inverted, and it can also have additional preamble-like data pre-pended.

45.3.1.2.1 Data Polarity

The Transmission Data that is received by the TX Digital module is mapped to a positive frequency deviation if the data is a 1, and is mapped to a negative frequency deviation if the data is a 0

This mapping can be reversed by setting the TX_CAPTURE_POL register bit.

45.3.1.2.2 Data Padding

Transmission Data Padding is done to avoid abrupt steps in On-Air frequency modulation and thereby minimize spectral transients during the transition from a low Power Amplifier setting to a high Power Amplifier setting prior to protocol data packet transmission.

The abrupt step in modulation is avoided by pre-pending symbols of preamble-like modulation onto the front end of the actual preamble transmission. The Power Amplifier ramp-up is done while modulating these additional symbols, and then the modulating power ramp-up smoothly transitions into the packet transmission at the Target Power level.

Data padding is supported for all of the Radio Protocols packet 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 first bit of the packet that follows. Data padding pre-pends the same 1 or 0 pattern as required to match the preamble case.

The data padding pattern is programmed into two 8-bit register fields to allow for two padding choices as seen in the BLE example shown above. If 16 bits of padding is selected, the 8-bit register fields are used twice for each case.

The DATA_PADDING_EN bits in the XCVR_TSM_CTRL register enable data padding as follows:

Table 45-6. Data Padding Selection

DATA_PADDING_EN[1:0]	Result
00	No Data Padding
01	8 bits or symbols of padding
10	16 bits or symbols of padding
11	Reserved

45.3.1.2.3 Power Amplifier Ramping

The XCVR_TSM provides the TX Digital module with the ramp-up timing needed to adjust the data padding to match the time when the start of the protocol data packet transmission needs to go On-Air.

During ramp-down of the Power Amplifier the TX Digital holds the TX Modulation Data at the final value until power reaches the minimum level.

Please refer to the section Power Amplifier Ramping in the Transceiver Sequence Manager chapter for more details on the Power Ramping topic.

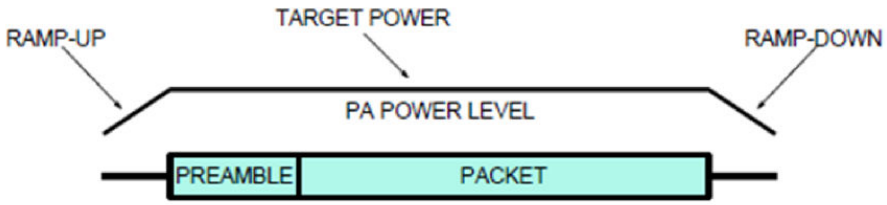


Figure 45-8. Power Amplifier Ramping

45.3.1.3 GFSK Modulator

45.3.1.3.1 Gaussian Filter

GFSK Modulation is implemented using a Gaussian filter and a lookup table to implement the various reference frequencies, data rates, and modulation index values.

The Gaussian filter is a symmetric 16-tap FIR filter with 11-bit quantized coefficients and supports reference frequencies of 32 and 26 MHz at oversampling ratios of 8 and 13 respectively. There are 2 sets of pre-programmed filter coefficients available as shown in the table below.

Table 45-7. Gaussian Filter Coefficients

Filter OSR	8	13
Tap Number(s)	32 MHz Ref	26 MHz Ref
1	1	23
2	4	41
3	13	68
4	41	103
5	99	144
6	192	186
7	300	220
8	374	239
9	374	239
10	300	220
11	192	186
12	99	144
13	41	103
14	13	68
15	4	41
16	1	23
Coefficient Sum	2048	2048

The pre-programmed filter coefficients can be manually overridden using the GFSK_FLD bit to disable the GFSK Filter Lookup Table. In this case the filter coefficients will be taken from the TX GFSK Filter Coefficients register.

45.3.1.3.2 Multiplier

The GFSK Modulator has a multiplier in hardware that uses a lookup table to choose the correct value to multiply the output of the Gaussian filter by in order to get the correct frequency deviation range.

The lookup table value chosen is dependent upon the Reference Clock frequency, the Transmitter Data Rate, and the Modulation Index.

The GFSK Multiplier Lookup Table value can be overridden by setting the GFSK_MLD register bit and programming the GFSK_MULTIPLY_TABLE_MANUAL register to contain a valid integer, using bits [15:4], and a valid fraction, using bits [3:0].

45.3.1.3.3 Modulation Scaling

The GFSK modulator supports nominal modulation indices of 0.32, 0.5, 0.7 and 1.0. Transmit modulation scaling adds the capability to scale the GFSK modulated output up or down by a factor 1/32, 1/16 or 1/8. The scaling coefficients have been chosen so that the scaling can be realized without a multiplication.

The TX modulation scaling is equivalent to having some programmability on the nominal modulation index, which can be useful to alter the modulation bandwidth and/or effective frequency deviation of the TX modulation signal.

Transmit modulation scaling is applied to the Gaussian Filter output.

Table 45-8. Transmit Modulation Scaling Choices

GFSK_MOD_INDEX_SCALING[2:0]	Modulation Scaling	Scaling of nominal modulation Index (%)
0	1	0
1	1 + 1/32	+3.125
2	1 + 1/16	+6.25
3	1 + 1/8	+12.5
4	1 - 1/32	-3.125
5	1 - 1/16	-6.25
6	1 - 1/8	-12.5

45.3.1.4 FSK Modulator

45.3.1.4.1 FSK Modulation Word

For 802.15.4 FSK Modulation, the FSK modulation word presented to the PLL is a direct mapping of a data 1 to the signed value of the FSK_MODULATION_SCALE_1 register, and of a data 0 to the signed value of the FSK_MODULATION_SCALE_0 register.

For Generic FSK, the FSK modulation word presented to the PLL is also scaled based on the BITRATE bits in the GENFSK register as shown in the table below.

This scaling can be disabled by setting the FSK_BITRATE_SCALE_DISABLE bit in the FSK_SCALE register.

Table 45-9. FSK Modulation Scaling

GENFSK BITRATE[1:0] Register	FSK Modulation Data Rate	FSK Modulation Scaling
0	1000 kbit/sec	Divide by 2
1	500 kbit/sec	Divide by 4
2	250 kbit/sec	Divide by 8
3	2000 kbit/sec	No Scaling

Note that a frequency inversion is allowed by inverting the values in the FSK_MODULATION_SCALE registers.

The default FSK Modulation mapping translates the chips/bits to a frequency modulation of +Frequency Deviation/PLL Minimum Modulation Step for a data 1, and -Frequency Deviation/PLL Minimum Modulation Step for a data 0.

For a 32 MHz reference clock with a PLL Minimum Modulation Step Size of 244.14 Hz, the default FSK Modulation mapping translates to a +/- 500 kHz peak Frequency Deviation, which is a PLL Baseband Frequency Word of +2047/-2047.

45.3.1.4.1.1 Half-Sine Pulse Shaping

The half-sine pulse shaping results from the inherent VCO reaction to the digital FSK modulation and the resulting modulation is as specified in IEEE 802.15.4 section 10.2.5, Figure 71, shown below.

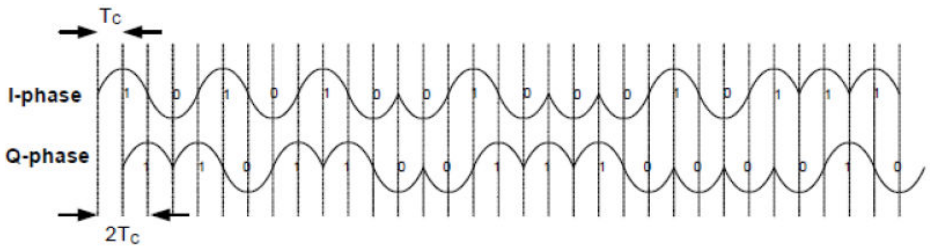


Figure 71—Sample baseband chip sequences with pulse shaping

Figure 45-9. Half-Sine Pulse Shaping

45.3.1.4.2 MSK Modes

Minimum-shift keying (MSK) is a type of continuous-phase frequency-shift keying with a frequency separation of one-half the bit rate. MSK modulation construction is quite similar to O-QPSK; MSK is encoded with bits alternating between quadrature components, with the Q component delayed by half the symbol period.

However, instead of square pulses as used by ordinary O-QPSK, MSK encodes each bit as a half sinusoid, which is similar to IEEE 802.15.4 O-QPSK. This results in a constant-modulus signal (constant envelope signal), which reduces problems caused by non-linear distortion.

In MSK the difference between the higher and lower frequency is identical to half the bit rate. Consequently, the waveforms used to represent a 0 and a 1 bit differ by exactly half a carrier period.

Thus, the maximum frequency deviation is $\delta = 0.25$ fm where fm is the maximum modulating frequency. As a result, the modulation index M is 0.5. This is the smallest FSK modulation index that can be chosen such that the waveforms for 0 and 1 are orthogonal.

As an example, 1 Mbps MSK would result in a frequency deviation of +/-250 kHz.

MSK data comes to the O-QPSK Modulator at the chip/data rate (no symbol to chip mapping is done), and the O-QPSK conversion is done on the data in the same manner as it is in IEEE 802.15.4-2011 section 10.2.5, as shown above.

45.3.1.5 Data Post-Processing

After the Transmission Data is processed by the GFSK or FSK Modulators, it can be post-processed to remove TX modulation images.

45.3.1.5.1 Sample/Hold Images

Power efficient constant envelope transmit modulation is done at a multiple of the symbol clock (Over-Sample Rate), which is chosen as a fraction of the RF Osc reference clock frequency. The transmit modulation signal is then converted to the reference clock rate, at which it is presented to the PLL as the modulation word. This sample/hold (or zero-order hold, ZOH) operation results in TX modulation images that repeat at the fractional reference clock rate at used.

These modulation images are automatically filtered by the sinc ZOH transfer function and the integration (i.e., 1/s) operation of the VCO. These inherent filtering operations result in these modulation images not violating any of the transmit modulation frequency mask. However, to provide additional spectral margin to support using an external power amplifier to boost the transmit power, an image suppression set of FIR filters has been included in the transmit path to suppress the transmit sample/hold images by at least an additional 20 dB.

The transmit image filters are enabled by default in GFSK and GMSK modes and not enabled in MSK modulation modes, but their usage can be overridden using the TX_IMAGE_FILTER_OVRD_EN bit.

45.3.1.5.2 Image Filters

The Transmitter uses up to three image filters, by default they are enabled based upon the modulation type and data rate as shown below:

Table 45-10. Default Transmit Image Filter Usage

Modulation Type; Data Rate	TX Image Filter 2	TX Image Filter 1	TX Image Filter 0
OQPSK/FSK/MSK Modes; All data rates	0	0	0
GFSK/GMSK; 1000 kbps	0	0	1
GFSK/GMSK; 500 kbps	0	1	1
GFSK/GMSK; 250 kbps	1	1	1

45.3.1.5.3 Image Filters Frequency Response

The frequency response of the TX image suppression filters and the reduction in the level of TX images for the 1 Mbps GMSK (BLE) with the FIR filter are demonstrated in the plots below.

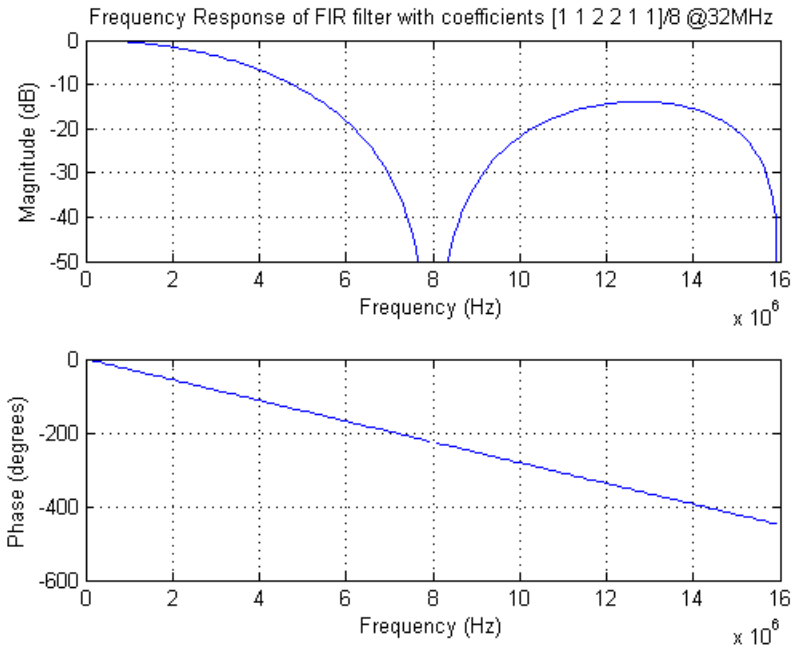


Figure 45-10. Frequency response of the 6-Tap FIR Transmit Image Suppression filter

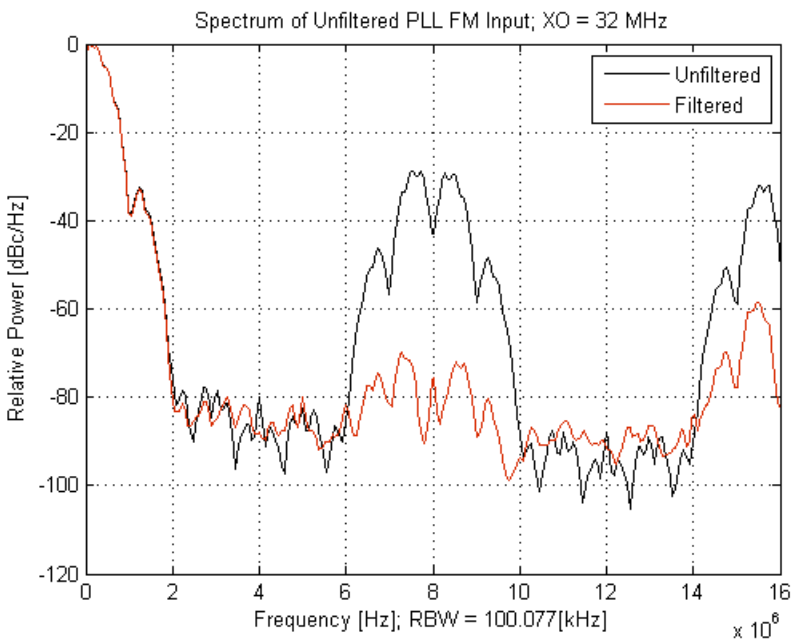


Figure 45-11. Comparison of Transmit Modulation Images at the PLL Input for BLE

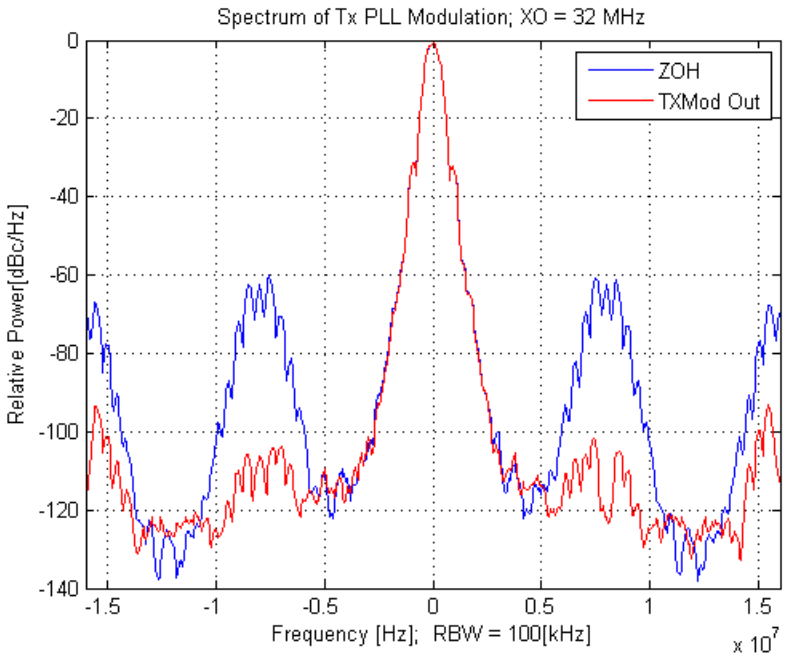


Figure 45-12. Comparison of Transmit Modulation Images at the PLL Output for BLE (1 Mbps GMSK)

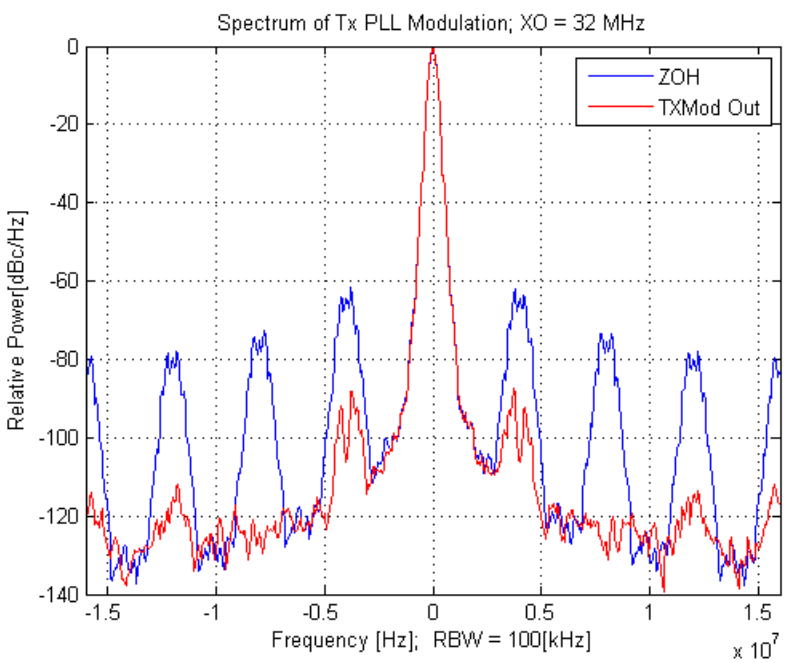


Figure 45-13. Comparison of Transmit Modulation Images at the PLL Output for 500 kbps GMSK

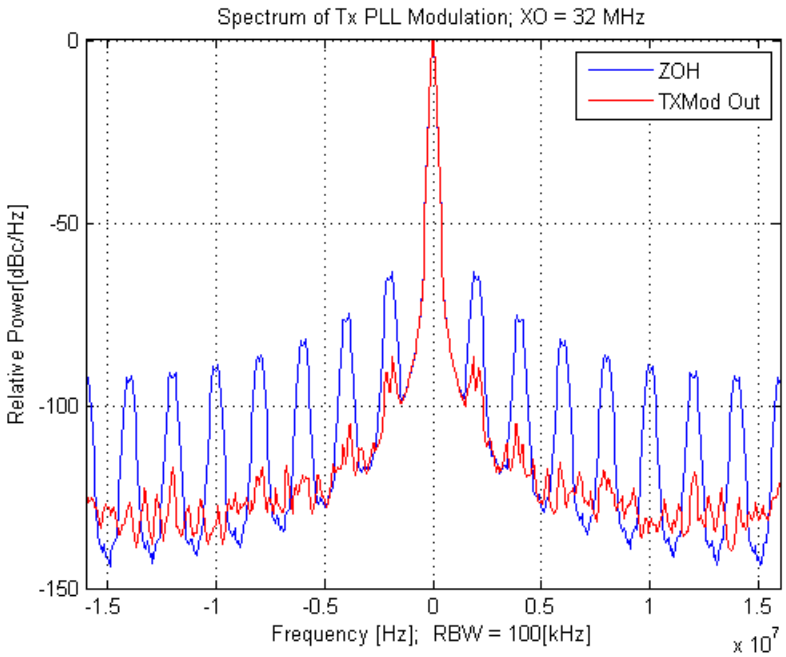


Figure 45-14. Comparison of Transmit Modulation Images at the PLL Output for 250 kbps GMSK

45.3.1.6 Memory Map and Register Definition

The Transmitter Digital memory map and detailed descriptions of all its registers are as follows.

45.3.1.6.1 XCVR_TX_DIG register descriptions

45.3.1.6.1.1 XCVR_TX_DIG_ADDR Memory map

XCVR_TX_DIG base address: 4005_C200h

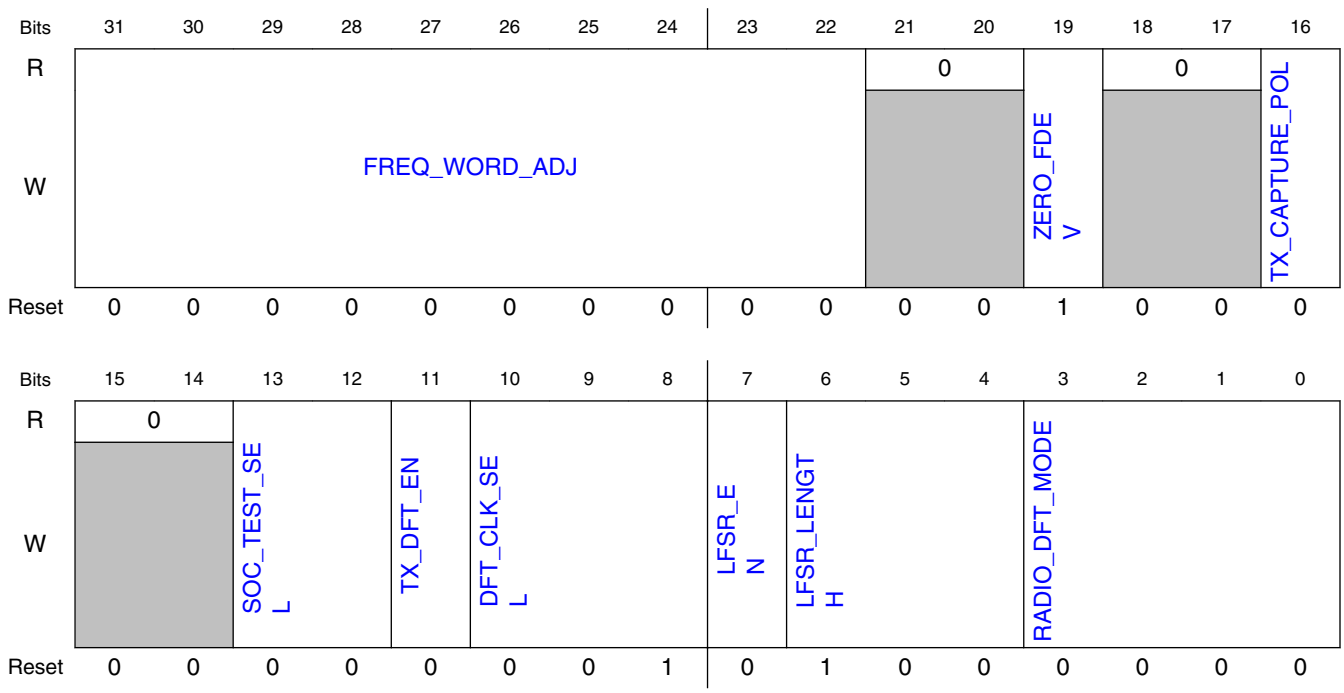
Offset	Register	Width (In bits)	Access	Reset value
0h	TX Digital Control (CTRL)	32	RW	0008_0140h
4h	TX Data Padding (DATA_PADDING)	32	RW	7FFF_55AAh
8h	TX GFSK Modulator Control (GFSK_CTRL)	32	RW	0001_4000h
Ch	TX GFSK Filter Coefficients 2 (GFSK_COEFF2)	32	RW	C063_0401h
10h	TX GFSK Filter Coefficients 1 (GFSK_COEFF1)	32	RW	BB29_960Dh
14h	TX FSK Modulation Levels (FSK_SCALE)	32	RW	0800_1800h
18h	TX DFT Modulation Pattern (DFT_PATTERN)	32	RW	0000_0000h

45.3.1.6.1.2 TX Digital Control (CTRL)

45.3.1.6.1.2.1 Offset

Register	Offset
CTRL	0h

45.3.1.6.1.2.2 Diagram



45.3.1.6.1.2.3 Fields

Field	Function
31-22 FREQ_WORD_ADJ	Frequency Word Adjustment This register is a signed 9 bit number that is added to the TX Digital output before it is presented to the PLL as the baseband frequency word. This allows the baseband frequency word to be adjusted, or skewed, by a range of -512 to +511. This frequency adjustment is applied to the final modulation word from any source (GFSK, FSK, DFT) and there is no protection from a math overflow. So the baseband frequency word range of -4096 to 4095 must be considered when adding this frequency word adjustment.
21-20 —	Reserved
19	On-Air at Zero FDev

Table continues on the next page...

Field	Function
ZERO_FDEV	If this bit is set, then in Gaussian Modulation modes the modulation will start at the Carrier Frequency instead of at the negative modulation value.
18-17 —	Reserved
16 TX_CAPTURE_POL	Polarity of the Input Data for the Transmitter If this bit is set, the TX data presented to the Transmitter will be inverted before it is processed.
15-14 —	Reserved
13-12 SOC_TEST_SEL	Radio Clock Selector for SoC RF Clock Tests This register selects the Radio clock source for the SoC Clock Tests Frequency Measurement. 00b - No Clock Selected 01b - PLL Sigma Delta Clock, divided by 2 10b - Auxiliary PLL Clock, divided by 2 11b - RF Ref Osc clock, divided by 2
11 TX_DFT_EN	DFT Modulation Enable If the Radio is in a DFT Pattern Register mode, then this bit is used to turn on and off the modulation that is shifted out from the pattern register.
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 in DFT Pattern Register modes, and the same frequency is also used to clock the LFSR and generate the pseudo-random modulation in DFT LFSR modes. 000b - 62.5 kHz 001b - 125 kHz 010b - 250 kHz 011b - 500 kHz 100b - 1 MHz 101b - 2 MHz 110b - 4 MHz 111b - RF OSC Clock
7 LFSR_EN	LFSR Enable If the Radio is in a DFT LFSR mode, then this bit is used to turn on and off the LFSR that is used to generate the modulation. Note that the LFSR is clocked at the DFT Clock frequency.
6-4 LFSR_LENGTH	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] 000b - LFSR 9, tap mask 100010000 001b - LFSR 10, tap mask 1001000000 010b - LFSR 11, tap mask 11101000000 011b - LFSR 13, tap mask 1101100000000 100b - LFSR 15, tap mask 111010000000000 101b - LFSR 17, tap mask 11110000000000000 110b - Reserved 111b - Reserved
3-0	Radio DFT Modes

Field	Function
RADIO_DFT_MODE	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. For LFSR modes the LFSR_EN needs to be set to turn on the LFSR. 0000b - Normal Radio Operation, DFT not engaged. 0001b - Carrier Frequency Only 0010b - Pattern Register GFSK 0011b - LFSR GFSK 0100b - Pattern Register FSK 0101b - LFSR FSK 0110b - Pattern Register O-QPSK 0111b - LFSR O-QPSK 1000b - LFSR 802.15.4 Symbols 1001b - PLL Modulation from RAM 1010b - PLL Coarse Tune BIST 1011b - PLL Frequency Synthesizer BIST 1100b - High Port DAC BIST 1101b - VCO Frequency Meter 1110b - Reserved 1111b - Reserved

45.3.1.6.1.3 TX Data Padding (DATA_PADDING)

45.3.1.6.1.3.1 Offset

Register	Offset
DATA_PADDING	4h

45.3.1.6.1.3.2 Diagram

Bits	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

Bits	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

45.3.1.6.1.3.3 Fields

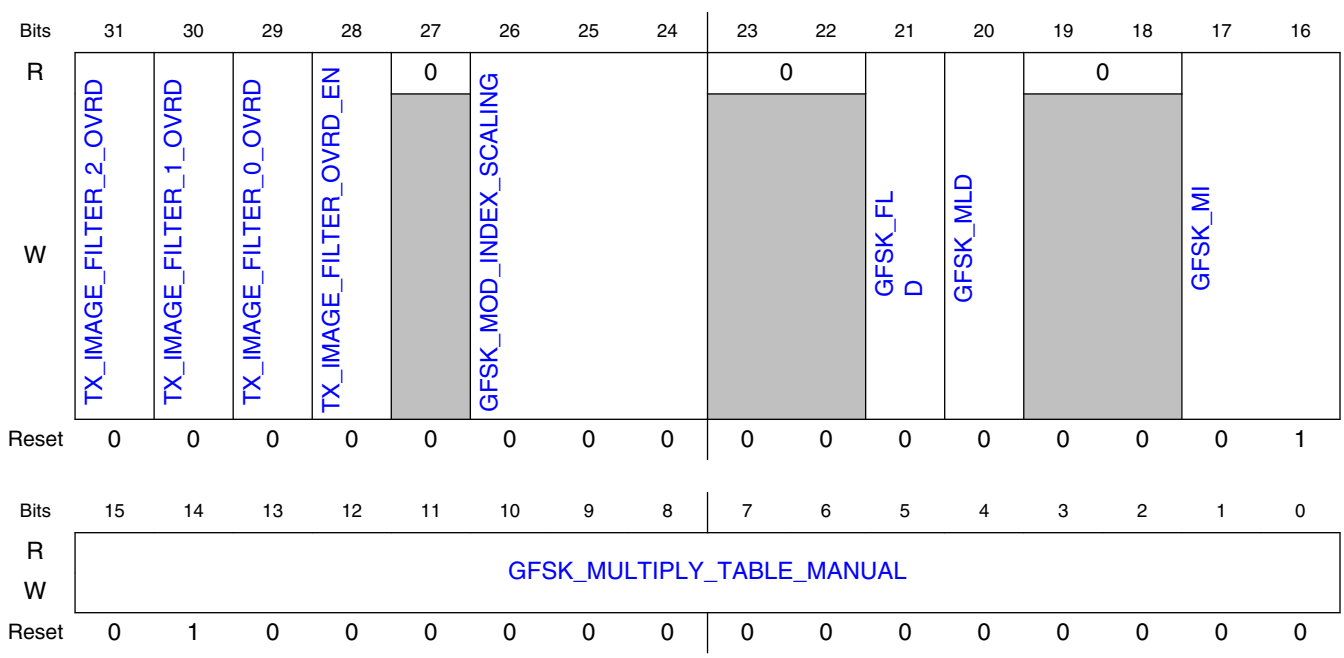
Field	Function
31 LRM	LFSR Reset Mask When this bit is set the DFT LFSR will not be reset when LFSR_EN is cleared and will instead continue to repeat its sequence as defined by the DFT_LFSR_LEN bits when LFSR_EN is next set. When this bit is cleared the DFT LFSR will reset every time LFSR_EN is cleared.
30-16 DFT_LFSR_OUT	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 the first bit of the Preamble is 1; the LSB is the first bit shifted out as padding.
7-0 DATA_PADDING_PAT_0	Data Padding Pattern 0 These bits are used for Data Padding when the first bit of the Preamble is 0; the LSB is the first bit shifted out as padding.

45.3.1.6.1.4 TX GFSK Modulator Control (GFSK_CTRL)

45.3.1.6.1.4.1 Offset

Register	Offset
GFSK_CTRL	8h

45.3.1.6.1.4.2 Diagram



45.3.1.6.1.4.3 Fields

Field	Function
31 TX_IMAGE_FILTER_2_OVRD	TX Image Filter 2 Override Control If the TX_IMAGE_FILTER_OVRD_EN bit is set, then this bit turns on (=1) and off (=0) the Transmit Image Filter 2
30 TX_IMAGE_FILTER_1_OVRD	TX Image Filter 1 Override Control If the TX_IMAGE_FILTER_OVRD_EN bit is set, then this bit turns on (=1) and off (=0) the Transmit Image Filter 1
29 TX_IMAGE_FILTER_0_OVRD	TX Image Filter 0 Override Control If the TX_IMAGE_FILTER_OVRD_EN bit is set, then this bit turns on (=1) and off (=0) the Transmit Image Filter 0
28 TX_IMAGE_FILTER_OVRD_EN	TX Image Filter Override Enable This bit enables the TX Image Filter Override Control bits.
27 —	Reserved
26-24 GFSK_MOD_INDEX_SCALING	GFSK Modulation Index Scaling Factor This register selects the amount to scale the Transmitter Modulation Word in GFSK Protocols. Transmit modulation scaling adds the capability to scale the GFSK modulated output up or down by a factor of 1/32, 1/16 or 1/8. This is equivalent to having some additional programmability of the modulation index. 000b - 1 001b - 1 + 1/32 010b - 1 + 1/16 011b - 1 + 1/8 100b - 1 - 1/32 101b - 1 - 1/16 110b - 1 - 1/8 111b - Reserved
23-22 —	Reserved
21 GFSK_FLD	Disable GFSK Filter Lookup Table 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.
20 GFSK_MLD	Disable GFSK Multiply Lookup Table If this bit is set, the GFSK Multiply Lookup table is disabled and GFSK_MULTIPLY_TABLE_MANUAL is used instead.
19-18 —	Reserved
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.

Table continues on the next page...

Field	Function
	The formula used for the Peak Modulation is { Symbol Rate / (2 x 1/Modulation Index) } 00b - 0.32 01b - 0.50 10b - 0.70 11b - 1.00
15-0 GFSK_MULTIPLY_TABLE_MANUAL	Manual GFSK Multiply Lookup Table 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] }

45.3.1.6.1.5 TX GFSK Filter Coefficients 2 (GFSK_COEFF2)

45.3.1.6.1.5.1 Offset

Register	Offset
GFSK_COEFF2	Ch

45.3.1.6.1.5.2 Function

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 the GFSK Manual Filter Coefficient [63:0]

45.3.1.6.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GFSK_FILTER_COEFF_MANUAL2															
Reset	1	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GFSK_FILTER_COEFF_MANUAL2															
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1

45.3.1.6.1.5.4 Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31-0
GFSK_FILTER_COEFF_MANUAL2</td><td>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.
<table>
<tr><th>Bits</th><th>Filter Coefficients</th></tr>
<tr><td>[36:32]</td><td>Filter coeff 0 and 15</td></tr>
<tr><td>[45:40]</td><td>Filter coeff 1 and 14</td></tr>
<tr><td>[55:48]</td><td>Filter coeff 4 and 11</td></tr>
<tr><td>[63:56]</td><td>Filter coeff 5 and 10</td></tr>
</table></td></tr>
</table>

45.3.1.6.1.6 TX GFSK Filter Coefficients 1 (GFSK_COEFF1)

45.3.1.6.1.6.1 Offset

Register	Offset
GFSK_COEFF1	10h

45.3.1.6.1.6.2 Function

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 the GFSK Manual Filter Coefficient [63:0]

45.3.1.6.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GFSK_FILTER_COEFF_MANUAL1															
Reset	1	0	1	1	1	0	1	1	0	0	1	0	1	0	0	1

Bits	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	0	1	0	1	1	0	0	0	0	0	1	1	0	1

45.3.1.6.1.6.4 Fields

Field	Function
31-0 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.

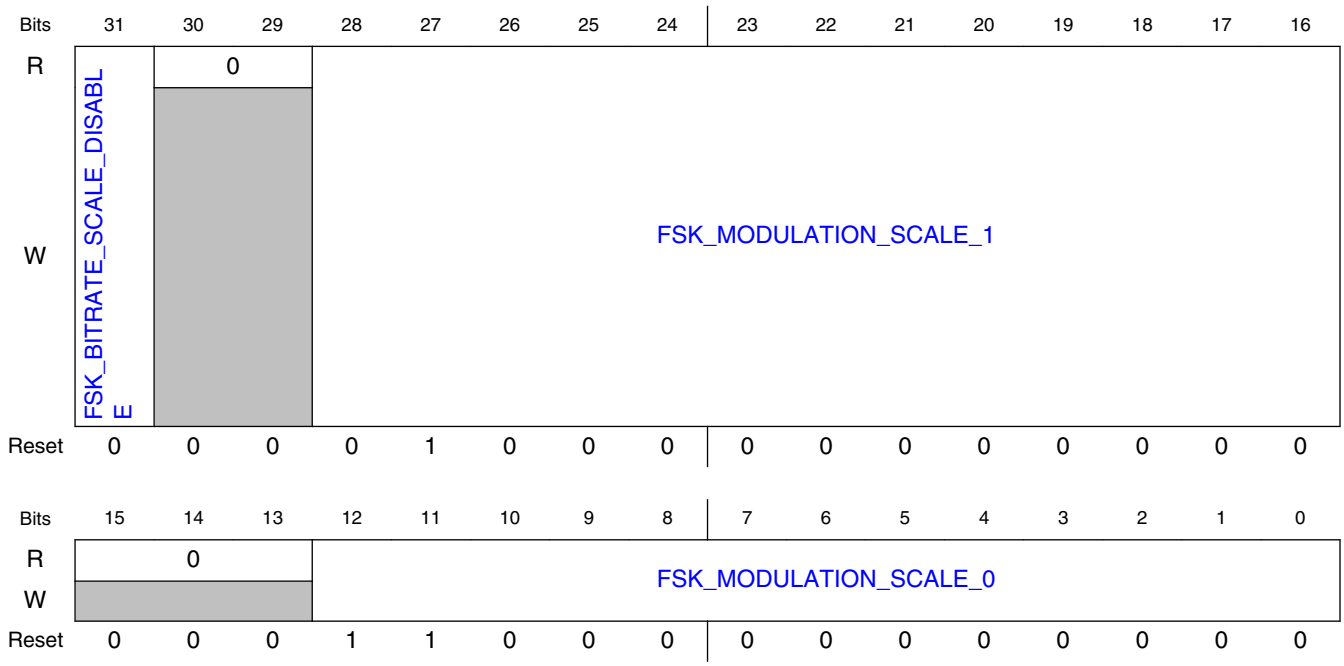
Bits	000 Filter Coefficients
[6:0]	Filter coeff 2 and 13
[15:7]	Filter coeff 6 and 9
[22:16]	Filter coeff 3 and 12
[31:23]	Filter coeff 7 and 8

45.3.1.6.1.7 TX FSK Modulation Levels (FSK_SCALE)

45.3.1.6.1.7.1 Offset

Register	Offset
FSK_SCALE	14h

45.3.1.6.1.7.2 Diagram



45.3.1.6.1.7.3 Fields

Field	Function
31 FSK_BITRATE_SCALE_DISABLE	FSK Bitrate Scaling Disable If this bit is set, the FSK Modulation values will not be scaled according to the bitrate setting.
30-29 —	Reserved
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 FSK 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 range of signed values that this register supports is -4096 to 4095
15-13 —	Reserved
12-0 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 FSK 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 range of signed values that this register supports is -4096 to 4095

45.3.1.6.1.8 TX DFT Modulation Pattern (DFT_PATTERN)

45.3.1.6.1.8.1 Offset

Register	Offset
DFT_PATTERN	18h

45.3.1.6.1.8.2 Diagram

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

45.3.1.6.1.8.3 Fields

Field	Function
31-0 DFT_MOD_PATTERN	DFT Modulation Pattern In TX DFT Pattern Register modes, if TX_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].

45.3.2 PLL Frequency Synthesizer

45.3.2.1 Introduction

45.3.2.1.1 About the PLL Frequency Synthesizer

The PLL Frequency Synthesizer is a Digital Module that -

1. Selects the Radio Carrier Frequency

2. Coarse Tunes the VCO and Calibrates the High Port DAC
3. Controls the PLL Loop Divider value to lock the VCO at the Carrier Frequency
4. Applies the Baseband Frequency Word as High Port Modulation (HPM) to the VCO High Port DAC
5. Applies the Baseband Frequency Word as Low Port Modulation (LPM) to the PLL Loop Divider
6. Monitors the PLL lock status and flags unlocked conditions
7. Provides DFT features for evaluating the PLL Frequency Synthesizer during factory validation and testing

45.3.2.1.2 Block diagram

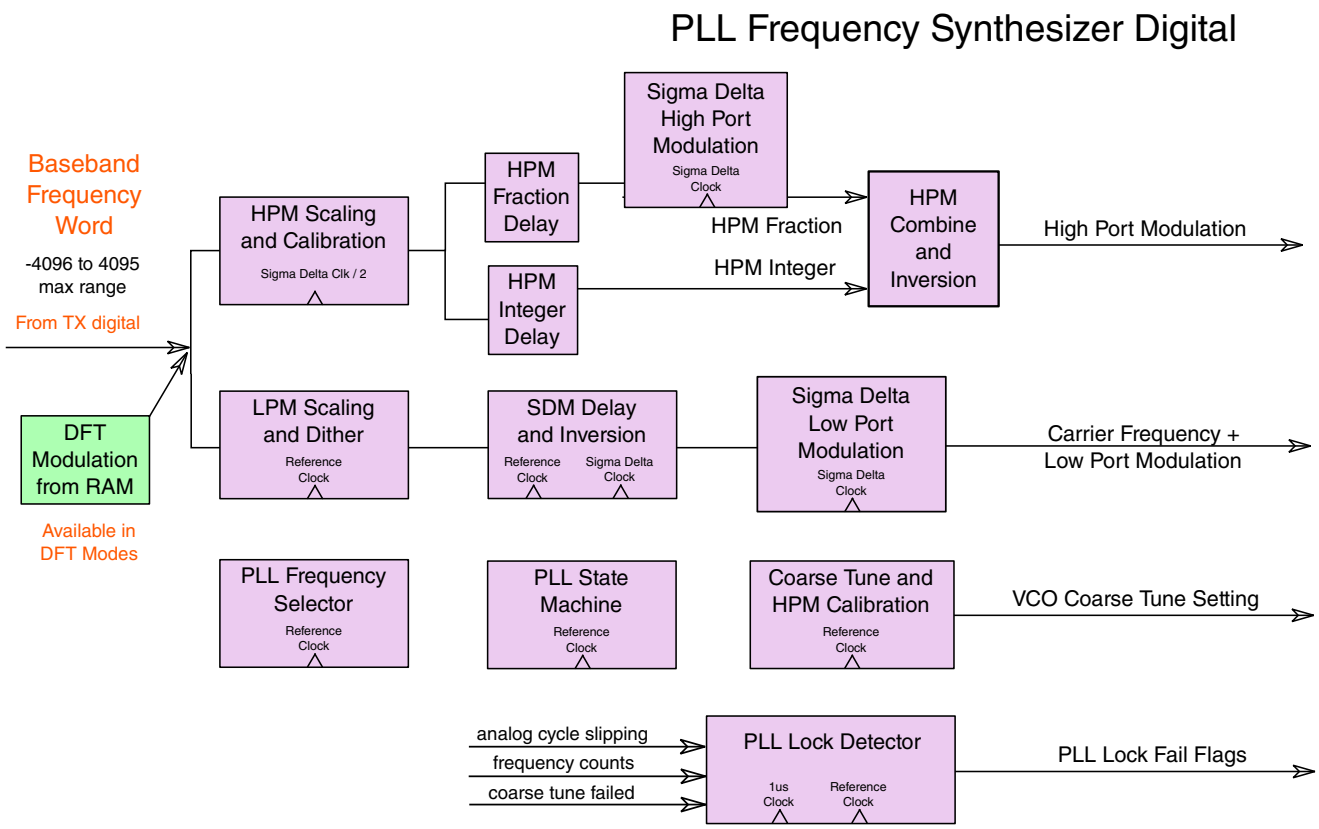


Figure 45-15. PLL Frequency Synthesizer

45.3.2.1.3 Baseband Frequency Word

The Transmitter Digital presents the PLL Frequency Synthesizer with a signed 13-bit Baseband Frequency word to modulate onto the RF Carrier Frequency. The Baseband Frequency Word range is -4096 to 4095, which provides a modulation range of +/- 1000 kHz when using a 32 MHz reference frequency with a PLL Minimum Frequency Step Size of 244.14 Hz.

45.3.2.1.3.1 Manual Frequency Word

The Baseband Frequency word can be overridden by software if the MOD_DISABLE bit is set. In this case the PLL Digital will use the MODULATION_WORD_MANUAL register as the source of the modulation.

Note that due to the speeds at which the High Port and Low Port react to changes in the modulation word, and the speed at which software is able to effect such changes, for all practical purposes this can only support low frequency modulation.

45.3.2.2 Carrier Frequency Tuning

45.3.2.2.1 Carrier Frequency Control

45.3.2.2.1.1 Radio Frequency Selection

The PLL Carrier Frequency is selected in these ways, depending on the Hop Configuration selected :

HOP_TBL_CFG	PLL Carrier Frequency Selection
0 or 1	Channel Mode The channel number, from the link layer or the Hop Frequency Word, is used by the Radio Frequency Selector to tune to a channel on the 1 MHz raster.
2	Frequency Offset Mode The link layer channel number, from the Hop Frequency Word, is used by the Radio Frequency Selector to tune to a channel on the 1 MHz raster. A signed 9-bit Frequency Offset from the Hop Frequency Word is added to the channel selected. This provides approximately a +/- 1 MHz adjustment to the 1 MHz raster channels. The step size of the adjustment is 3.906 kHz for a 32 MHz clock and 3.174 kHz for a 26 MHz clock.
3	Direct Carrier Frequency Selection An Integer is selected and the Hop Frequency Word is applied directly as a signed 15-bit value for the Numerator. This provides tuning with approximately a 2 kHz frequency resolution that covers the full 2400 - 2480 MHz range with a 32 MHz clock, and covers a range of 2400 - 2470 MHz with a 26 MHz clock.
4,5,6,7	Reserved

The Radio mapping of the 128 RF channels on the 1 MHz raster, spanning the range from 2.360 GHz to 2.487 GHz, is shown in the figure below along with the Hop Frequency Word options.

There is also an option to allow an override of the Protocol Channel Frequency Mapping in order to allow any Protocol to choose to use any of the available Channel Frequency Mapping tables.

Note that there is also a full manual mode where the Integer and Fraction can be selected by software and thereby choose any RF Carrier Frequency in increments of the PLL SDM LSB (as long as the requested frequency is within the VCO specified range)

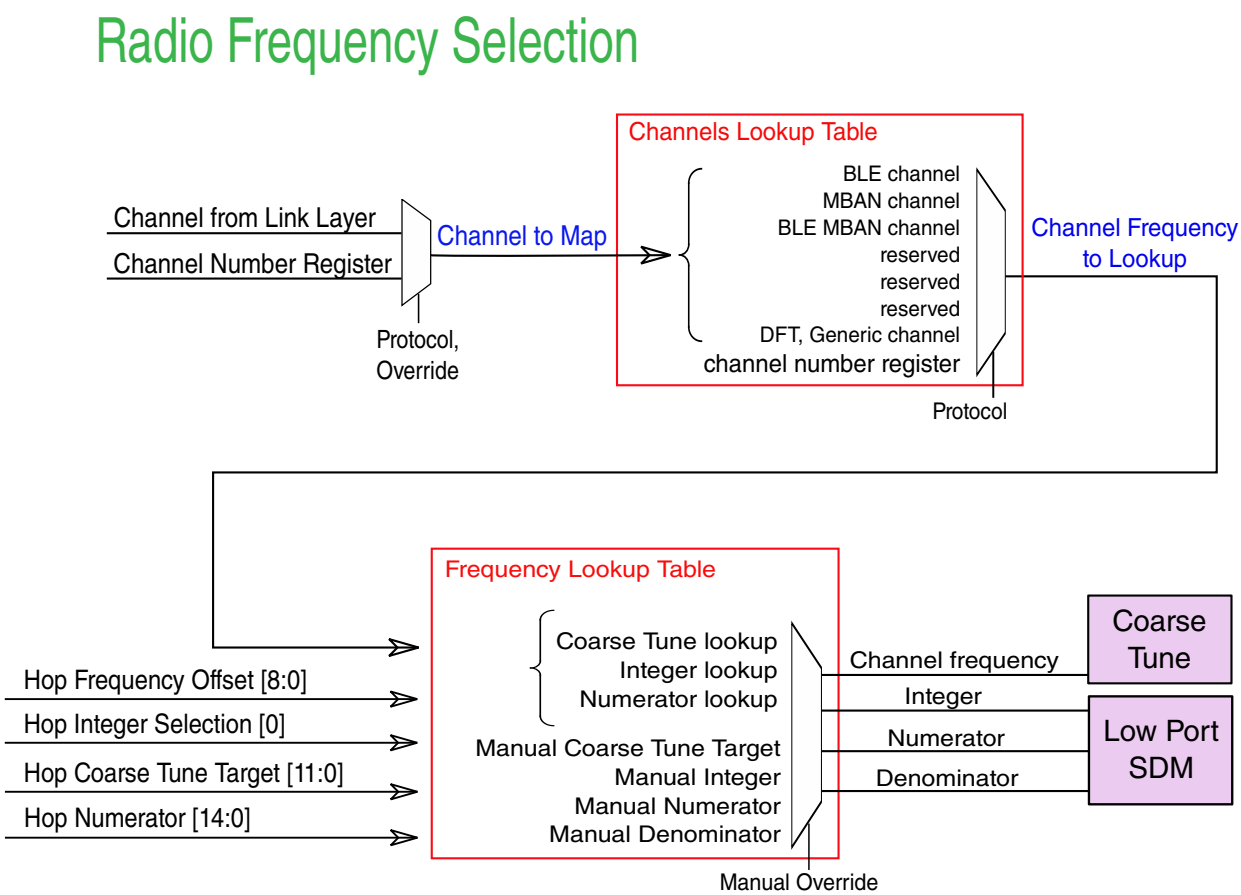


Figure 45-16. Radio Frequency Selection

45.3.2.2.1.2 RF Channels Supported

The channel frequencies for each protocol are shown in the tables below.

Table 45-11. Radio Protocols (0 to 2) Channel Frequencies

Radio Protocols Supported, Channel Frequency in MHz					
BLE		MBAN		BLE/MBAN	
0		1		2	
Chan	MHz	Chan	MHz	Chan	MHz
0	2,402	0	2,360	0	2,402
1	2,404	1	2,361	1	2,404
2	2,406	2	2,362	2	2,406
3	2,408	3	2,363	3	2,408
4	2,410	4	2,364	4	2,410
5	2,412	5	2,365	5	2,412
6	2,414	6	2,366	6	2,414
7	2,416	7	2,367	7	2,416
8	2,418	8	2,368	8	2,418
9	2,420	9	2,369	9	2,420
10	2,422	10	2,370	10	2,422
11	2,424	11	2,371	11	2,424
12	2,426	12	2,372	12	2,426
13	2,428	13	2,373	13	2,428
14	2,430	14	2,374	14	2,430
15	2,432	15	2,375	15	2,432
16	2,434	16	2,376	16	2,434
17	2,436	17	2,377	17	2,436
18	2,438	18	2,378	18	2,438
19	2,440	19	2,379	19	2,440
20	2,442	20	2,380	20	2,442
21	2,444	21	2,381	21	2,444
22	2,446	22	2,382	22	2,446
23	2,448	23	2,383	23	2,448
24	2,450	24	2,384	24	2,450
25	2,452	25	2,385	25	2,452
26	2,454	26	2,386	26	2,454
27	2,456	27	2,387	27	2,456
28	2,458	28	2,388	28	2,458
29	2,460	29	2,389	29	2,460
30	2,462	30	2,390	30	2,390
31	2,464	31	2,391	31	2,391
32	2,466	32	2,392	32	2,392

Table continues on the next page...

Table 45-11. Radio Protocols (0 to 2) Channel Frequencies (continued)

33	2,468	33	2,393	33	2,393
34	2,470	34	2,394	34	2,394
35	2,472	35	2,395	35	2,395
36	2,474	36	2,396	36	2,396
37	2,476	37	2,397	37	2,397
38	2,478	38	2,398	38	2,398
39	2,480	39	2,399	39*	2,480

* The BLE MBAN Channel Remap bit (BMR) controls the frequency mapping for channel 39 in BLE/MBAN, 2480 MHz is the default, but this channel can be remapped to 2399 MHz by setting BMR to a 1.

Table 45-12. Radio Protocols (6 to 9) Channel Frequencies

Radio Protocols Supported, Channel Frequency in MHz					
DFT, Generic					
6,7,8,9					
Channel	MHz	Channel	MHz	Channel	MHz
0	2,360	43	2,403	86	2,446
1	2,361	44	2,404	87	2,447
2	2,362	45	2,405	88	2,448
3	2,363	46	2,406	89	2,449
4	2,364	47	2,407	90	2,450
5	2,365	48	2,408	91	2,451
6	2,366	49	2,409	92	2,452
7	2,367	50	2,410	93	2,453
8	2,368	51	2,411	94	2,454
9	2,369	52	2,412	95	2,455
10	2,370	53	2,413	96	2,456
11	2,371	54	2,414	97	2,457
12	2,372	55	2,415	98	2,458
13	2,373	56	2,416	99	2,459
14	2,374	57	2,417	100	2,460
15	2,375	58	2,418	101	2,461
16	2,376	59	2,419	102	2,462
17	2,377	60	2,420	103	2,463
18	2,378	61	2,421	104	2,464
19	2,379	62	2,422	105	2,465
20	2,380	63	2,423	106	2,466
21	2,381	64	2,424	107	2,467

Table continues on the next page...

Table 45-12. Radio Protocols (6 to 9) Channel Frequencies (continued)

22	2,382	65	2,425	108	2,468
23	2,383	66	2,426	109	2,469
24	2,384	67	2,427	110	2,470
25	2,385	68	2,428	111	2,471
26	2,386	69	2,429	112	2,472
27	2,387	70	2,430	113	2,473
28	2,388	71	2,431	114	2,474
29	2,389	72	2,432	115	2,475
30	2,390	73	2,433	116	2,476
31	2,391	74	2,434	117	2,477
32	2,392	75	2,435	118	2,478
33	2,393	76	2,436	119	2,479
34	2,394	77	2,437	120	2,480
35	2,395	78	2,438	121	2,481
36	2,396	79	2,439	122	2,482
37	2,397	80	2,440	123	2,483
38	2,398	81	2,441	124	2,484
39	2,399	82	2,442	125	2,485
40	2,400	83	2,443	126	2,486
41	2,401	84	2,444	127	2,487
42	2,402	85	2,445		

45.3.2.2.1.3 Channel Selector

During typical Radio usage, the channel selection is done automatically by the Link Layer.

Channel selection can be overridden using the PLL_CHAN_MAP register bit BOC (BLE protocols 0 and 2). If this bit set then the CHANNEL_NUM register selects the Radio Channel Number instead of the Link Layer.

45.3.2.2.1.4 Manual Carrier Frequency

The internal frequency mapping can be bypassed and software can directly control the Low Port Sigma Delta inputs to select any channel frequency in the VCO's range (in increments of the Low Port Minimum Frequency Step Size).

The Radio Channel Frequency override is manually selected by setting the SDM_MAP_DISABLE bit along with the LPM_INTG, LPM_NUM, and LPM_DENOM register values.

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.

45.3.2.2.2 Hop Frequency Word

The Hop Frequency Word is used to provide the PLL with Channel Selection, Frequency Offset, Integer Selection, and Direct Numerator Control of the Low Port Sigma Delta Modulator.

The figure below shows an overview of how the Hop Table interfaces to the PLL.

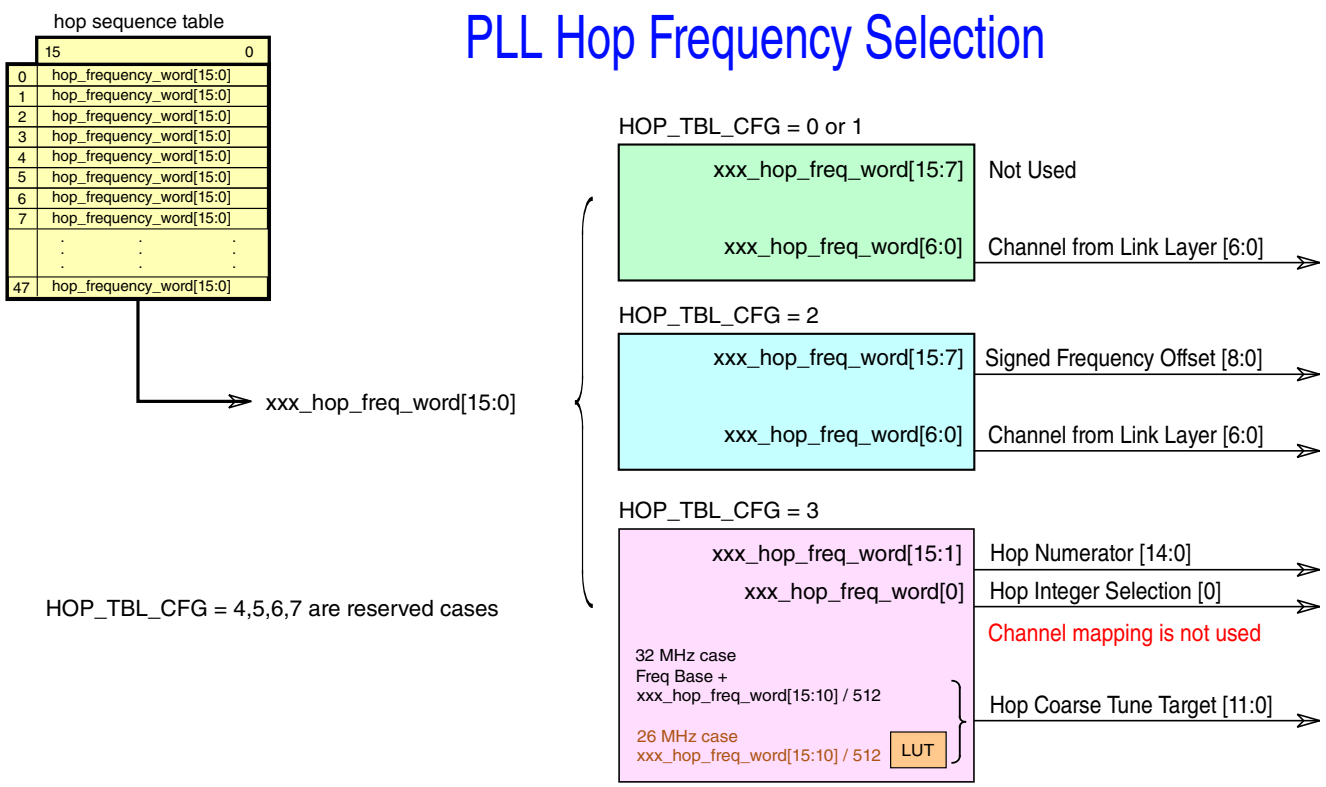


Figure 45-17. PLL Hop Frequency Selection

For much of the Radio usage the mapping of the 128 RF channels on a 1 MHz raster is sufficient. However, there are additional cases where Carrier Frequency Tuning at a finer resolution is desired and the Hop Frequency Word contains the information needed to get these finer resolutions.

The table below shows some of these desired Channel Spacings.

Modulation Type	Data Rate	Sensitivity Estimate (dBm)	Frequency Deviation (kHz)	99.5% Signal BW (kHz)	Typical Channel Spacing (kHz)	Notes	Desired Offsets from 1 MHz Channel Map
GFSK BT=0.5, h=0.5	1000	-96	250	1105	2000	Bluetooth-LE	1000
	500	-98	125	552.5	1000		0
	250	-100	62.5	276.3	500	Gen FSK 4	500
GFSK, BT=0.5, h= 0.32	1000	-92	160	926	1000	ANT	0
	500	-94	80	463	600		400
	250	-96	40	231.5	400		600
GFSK, BT=0.5, h=0.7	1000	-96	350	1278	2000		1000
	500	-99	175	639	1000	Gen FSK 1	0
	250	-101	87.5	319.5	500		500
GFSK, BT=0.5, h=1.0	1000	-93	500	1765	3000		2000
	500	-96	250	882.5	1500		500
	250	-98	125	441.3	750	Gen FSK 5	250
GMSK BT=0.3	1000	-93	250	975	1200	Gen FSK 2	200
	500	-95	125	487.5	600		400
	250	-97	62.5	243.8	400		600
GMSK BT=0.7	1000	-96	250	1167	2000		1000
	500	-98	125	583.5	1000		0
	250	-100	62.5	291.8	500		500
MSK/FSK (h=0.5, half-sin)	1000	-93	250	1605	2500		1500
	500	-95	125	802.5	1250	Gen FSK 3	250
	500	-95	125	802.5	1300	Gen FSK 4	300
	250	-98	62.5	401.3	625		375
	250	-98	62.5	401.3	650		350

Figure 45-18. PLL Typical Channel Frequency Spacings

45.3.2.2.2.1 Frequency Offset Mode

In order to support the desired Channel Spacings, a Frequency Offset Mode is provided that allows the Carrier Frequency to be offset from the selected channel in approximately 4 kHz steps. The tables below shows the resulting Frequency Error for some of the channel spacings.

The Hop Frequency Word contains a 1 MHz channel number and a 9-bit signed offset value. The offset is left shifted and then applied to the Numerator from the channel lookup table to create the Numerator applied to the Low Port SDM.

The Frequency Offset Range is approximately +/- 1 MHz

The Coarse Tune Target is not effected in this mode.

Integer	Min Fraction	Max Fraction	SDM Denominator	Min Numerator	Max Numerator	Min Frequency	Max Frequency	Frequency Range
38	-0.5	0.5	1,048,576	-16,320	16,320	2,431,003,906	2,432,996,094	1,992,188

Fref	32	MHz
Denom_resolution	20	bits
Denominator	1,048,576	
Channel bits	7	
Integer bits	0	
Hop numerator width, plus sign bit	8	
Left shift Hop Num	6	
min numerator value	1	
min/max signed numerator value	255	

	Numerator Value	Numerator Shifted Value	Frequency Result	Desired Frequency	Frequency Error
	1	64	3,906		
	8	512	31,250		
	16	1,024	62,500		
	32	2,048	125,000		
Desired Offsets from 1 MHz Channel Map	51	3,264	199,219	200,000	-0.3906%
	64	4,096	250,000	250,000	0.0000%
	77	4,928	300,781	300,000	0.2604%
	90	5,760	351,563	350,000	0.4464%
	96	6,144	375,000	375,000	0.0000%
	102	6,528	398,438	400,000	-0.3906%
	128	8,192	500,000	500,000	0.0000%
	154	9,856	601,563	600,000	0.2604%
	160	10,240	625,000		
	192	12,288	750,000		
	224	14,336	875,000		
	240	15,360	937,500		
	255	16,320	996,094		

Figure 45-19. PLL Hop Frequency Offset Mode, 32 MHz

Integer	Min Fraction	Max Fraction	SDM Denominator	Min Numerator	Max Numerator	Min Frequency	Max Frequency	Frequency Range
46	-0.5	0.5	1,048,576	-16,320	16,320	2,391,190,674	2,392,809,326	1,618,652

Fref	26	MHz
Denom_resolution	20	bits
Denominator	1,048,576	
Channel bits	7	
Integer bits	0	
Hop numerator width, plus sign bit	8	
Left shift Hop Num	6	
min numerator value	1	
min/max signed numerator value	255	

	Numerator Value	Numerator Shifted Value	Frequency Result	Desired Frequency	Frequency Error
	1	64	3,174		
	8	512	25,391		
	16	1,024	50,781		
	32	2,048	101,563		
Desired Offsets from 1 MHz Channel Map	63	4,032	199,951	200,000	-0.0244%
	79	5,056	250,732	250,000	0.2930%
	95	6,080	301,514	300,000	0.5046%
	110	7,040	349,121	350,000	-0.2511%
	118	7,552	374,512	375,000	-0.1302%
	126	8,064	399,902	400,000	-0.0244%
	158	10,112	501,465	500,000	0.2930%
	189	12,096	599,854	600,000	-0.0244%
	160	10,240	507,813		
	192	12,288	609,375		
	224	14,336	710,938		
	240	15,360	761,719		
	255	16,320	809,326		

Figure 45-20. PLL Hop Frequency Offset Mode, 26 MHz

45.3.2.2.2.2 Direct Carrier Frequency Mode

For a finer Carrier Frequency resolution, the Direct Carrier Frequency mode is provided which allows approximately a 2 kHz step size. This method allows for complete coverage of the 2400 - 2480 MHz range while using a 32 MHz reference clock, and a range of 2400 - 2470 MHz while using a 26 MHz reference clock. The Frequency Errors from a 2 kHz raster are shown in the tables below.

The Hop Frequency Word contains the Integer choice and a 15-bit signed value to be left shifted and then applied as the Numerator to the Low Port SDM.

The Frequency Offset Range is +/- 32 MHz for a 32 MHz clock, and +/- 26 MHz for a 26 MHz clock.

In this mode the Coarse Tune Target is specially calculated as follows :

- For 32 MHz clock operations, the signed 15-bit Hop Frequency Word is divided by 512 and added to the selected Frequency Base.
  - Frequency Base for integer 38 is 2432, for integer 39 it is 2496
- For 26 MHz clock operations, the signed 15-bit Hop Frequency Word is divided by 512 and then a Look Up Table is used to select the best Coarse Tune Target.

Integer	Min Fraction	Max Fraction	SDM Denominator	Min Numerator	Max Numerator	Min Frequency	Max Frequency	Frequency Range
38	-0.5	0.5	1,048,576	-524,288	524,256	2,400,000,000	2,463,998,047	63,998,047
39	-0.5	0.5	1,048,576	-524,288	524,256	2,464,000,000	2,527,998,047	63,998,047

Fref	32	MHz
Denom_resolution	20	bits
Denominator	1,048,576	
Channel bits	0	
Integer bits	1	(38,39) or (46, 47)
Hop numerator width, plus sign bit	14	

Left shift Hop Num by 5	2 kHz Freq Steps	Numerator Value	Numerator Shifted Value	Frequency Result	Frequency Error
	2,000	1	32	1,953	-2.3438%
	4,000	2	64	3,906	-2.3438%
	6,000	3	96	5,859	-2.3438%
	8,000	4	128	7,813	-2.3438%
	42,000	22	704	42,969	2.3065%
	44,000	23	736	44,922	2.0952%
	46,000	24	768	46,875	1.9022%
	48,000	25	800	48,828	1.7253%
	62,000	32	1024	62,500	0.8065%
	64,000	33	1056	64,453	0.7080%
	66,000	34	1088	66,406	0.6155%
	68,000	35	1120	68,359	0.5285%
	31,992,000	16380	524160	31,992,188	0.0006%
	31,994,000	16381	524192	31,994,141	0.0004%
	31,996,000	16382	524224	31,996,094	0.0003%
	31,998,000	16383	524256	31,998,047	0.0001%
	32,000,000	16384	524288	32,000,000	0.0000%
Max Positive Value		16383			

Figure 45-21. PLL Hop Direct Carrier Frequency Selection, 32 MHz

Integer	Min Fraction	Max Fraction	SDM Denominator	Min Numerator	Max Numerator	Min Frequency	Max Frequency	Frequency Range
46	-0.5	0.5	1,048,576	-524,288	524,256	2,366,000,000	2,417,998,413	51,998,413
47	-0.5	0.5	1,048,576	-524,288	524,256	2,418,000,000	2,469,998,413	51,998,413

Note that we would need to push the fraction on the LP SDM to cover the range above 2470, or add a bit somewhere

Fref	26	MHz
Denom_resolution	20	bits
Denominator	1,048,576	
Channel bits	0	
Integer bits	1	(38,39) or (46, 47)
Hop numerator width, plus sign bit	14	

Left shift Hop Num by 5	2 kHz Freq Steps	Numerator Value	Numerator Shifted Value	Frequency Result	Frequency Error
	2,000	1	32	1,587	-20.6543%
	4,000	3	96	4,761	19.0186%
	6,000	4	128	6,348	5.7943%
	8,000	5	160	7,935	-0.8179%
	42,000	26	832	41,260	-1.7625%
	44,000	28	896	44,434	0.9854%
	46,000	29	928	46,021	0.0446%
	48,000	30	960	47,607	-0.8179%
	62,000	39	1248	61,890	-0.1780%
	64,000	40	1280	63,477	-0.8179%
	66,000	42	1344	66,650	0.9854%
	68,000	43	1376	68,237	0.3490%
	25,992,000	16379	524128	25,992,065	0.0003%
	25,994,000	16380	524160	25,993,652	-0.0013%
	25,996,000	16381	524192	25,995,239	-0.0029%
	25,998,000	16383	524256	25,998,413	0.0016%
	26,000,000	16384	524288	26,000,000	0.0000%
Max Positive Value		16383			

Figure 45-22. PLL Hop Direct Carrier Frequency Selection, 26 MHz

45.3.2.3 VCO Calibration

45.3.2.3.1 Coarse Tune

The VCO Coarse Tune is carried out by a Successive Approximation Register method that computes the best coarse tune setting for the VCO in successive steps from MSB to LSB.

This SAR method uses a Frequency Meter count value that is successively compared to expected counts to determine the best coarse tune setting for the VCO.

The coarse tune setting can be overridden for test and validation purposes, and the individual frequency counts used at each successive step for the coarse tune setting are available for viewing in DTEST mode and also as register values in the PLL register space.

45.3.2.3.2 High Port DAC Calibration

The objective of the HPM DAC Calibration is to determine the DAC Step Size and estimate the ratio of this DAC unit step to the Low Port Minimum Frequency Step Size.

Once the step size of the High Port DAC is known, a scaling factor is applied to correctly map the baseband frequency word to the High Port modulation.

The length of the calibration time is set using the HPM_CAL_TIME register and the size of the DAC array used is selected using the HPM_CAL_ARRAY_SIZE register.

The register HPM_COUNT_ADJUST[3:0] provides a (-8/+7) range of adjustment to the HPM Calibration Count Difference for the HPM Calibration lookup table to allow for any count error bias that may be present. This adjusted Count Difference is used with a Lookup Table to choose the HPM Calibration Factor

45.3.2.3.2.1 HPM CAL Factor

The HPM_CAL_FACTOR register shows the value that is currently being used by High Port Modulation Multiplier. This value is from the most recent High Port calibration unless it is being overridden using the HP_CAL_DISABLE and HPM_CAL_FACTOR_MANUAL registers.

45.3.2.4 High Port Modulation

The High Port Modulator multiplies the HPM Calibration Factor and the PLL Baseband Frequency Word to get the High Port Modulation Word.

The High Port Modulation Word is represented by both an Integer and a Fractional portion. The Fraction is processed by the High Port Sigma Delta Modulator and the result is combined with the Integer and used to control the VCO High Port DAC.

45.3.2.4.1 High Port DAC

The High Port Digital to Analog Converter is designed to take the High Port Modulation Word and convert it to a control voltage that is applied as modulation to the VCO High Port.

45.3.2.4.1.1 HPM DAC Settings

The HPM DAC settings during calibration and normal operation are shown in the table below.

Warning : TX operation with On-Air modulation ranges greater than +/- 250 kHz (+/- 500 kHz at the VCO) require that the VCO Kvm be adjusted using the SY_VCO_KVM[2:0] register.

Table 45-13. Nominal HPM DAC parameters in various modes

Radio TX Mode		DAC max swing (mV)	DAC output common mode (mV)	DAC swing during operation	Vcm (mV) / HPM_VCM	OPAMP gain setting (R2/R1) / HPM_FDB_RES	Common mode at varactor (mV)	Swing at Varactor input (mV)	Kvm (MHz/Kv)	Deviation at VCO (KHz)[1]
2000kHz	HPM calibration	+/-82	82	+/-82	288 / 3'b001	2/ 2'b01	700	+/-164	20	+/-3280
	TX operation	+/-82	82	+/-51.2	391 / 3'b000	1/ 2'b00	700	+/51.2	20	+/-1024
BLE/ 1000kHz	HPM calibration	+/-82	82	+/-82	288 / 3'b001	2 / 2'b01	700	+/-164	10	+/-1640
	TX operation	+/-82	82	+/-51.2	391 / 3'b000	1/ 2'b00	700	+/51.2	10	+/-512
500kHz	HPM calibration	+/-82	82	+/-82	288 / 3'b001	2/ 2'b01	700	+/-164	10	+/-1640
	TX operation	+/-82	82	+/-25.6	391 / 3'b000	1/ 2'b00	700	+/25.6	10	+/-256
250kHz	HPM calibration	+/-82	82	+/-82	288 / 3'b001	2/ 2'b01	700	+/-164	10	+/-1640
	TX operation	+/-82	82	+/-12.8	391 / 3'b000	1/ 2'b00	700	+/12.8	10	+/-128

1. On-Air modulation frequency is a divide by 2 of the VCO frequency.

45.3.2.4.1.2 HPM DAC Bump

The HPM DAC Calibration is normally done at 2X the On-Air Modulation Range, this is accomplished by writing the PLL HPM Analog Bump Control (PLL_HPM_BUMP) register bits:

- HPM_VCM_CAL, value applied during HPM DAC Calibration
- HPM_FDB_RES_CAL, value applied during HPM DAC Calibration
- HPM_VCM_TX, value applied during Radio Transmissions
- HPM_FDB_RES_TX, value applied during Radio Transmissions

If the registers are written in such a way that the DAC range is not doubled, then the HPM_CAL_NOT_BUMPED bit must be set to change the math in the HPM Calibration.

45.3.2.4.1.3 HPM DAC 1 MHz Range

TX operation with On-Air modulation ranges greater than +/- 250 kHz (+/- 500 kHz at the VCO) require that the VCO Kvm be adjusted using the SY_VCO_KVM[2:0] register.

This VCO Kvm setting should be made prior to HPM DAC calibration.

45.3.2.4.2 High Port Sigma Delta

The High Port fractional modulation is implemented using a 2nd-order MASH Sigma Delta Modulator (SDM), which converts the 6-bit fractional portion of the high port modulation word to a 3-state thermometric code {-1,0,+1}. The resulting code is then added to the integer portion of the high port modulation word and the final result is applied to the HPM DAC.

45.3.2.4.2.1 HPM SDM Dither

In addition to the nominal SDM functionality, the HPM SDM design includes a zero-mean dithering to mitigate idle tones that can occur when modulating a string of ones or zeros.

The Dither is implemented using a Linear-Feedback Shift Register (LFSR) which provides a 1-bit pseudo-randomization that adds or subtracts 1-bit to the 6-bit fractional portion before it is processed by the Sigma Delta Modulator.

The length of the LFSR can be selected using the HPM_LFSR_SIZE register, and the scale can be changed using the HPM_DTH_SCL bit. The Dither can also be disabled by clearing the HPM_DTH_EN bit.

45.3.2.4.3 High Port Delay, Disable, Invert

The HPM Integer and Fraction portions each have an array of programmable delays available to allow the HPM modulations to be matched to each other and to the LPM modulation and the delay through the PLL Loop Divider and onto the VCO feedback loop.

The HPM delay steps are in increments of the PLL Sigma Delta Clock divided by 2, and the LPM SDM delay steps are in increments of the PLL Sigma Delta Clock. All three delay step settings are in the range of {0 to 15}.

There are several inversion bits that can be used if needed:

- HPM_CAL_INVERT, Invert the High Port Modulator during Calibration
- HPM_MOD_IN_INVERT, Invert the input word to the High Port Modulator
- HPM_INTEGER_INVERT, Invert the High Port Modulation Integer portion
- HPM_SDM_OUT_INVERT, Invert the HPM SDM Fractional Output

High Port Modulation can be disabled by setting the HPM_MOD_DISABLE bit, in this case the HPM_MOD_MANUAL register value is applied to the VCO High Port.

45.3.2.5 Low Port Modulation

The Low Port Modulator uses a 3rd order Sigma-Delta to produce a fractional division (Fractional-N Frequency Synthesis) to control the PLL Loop Divider and tune the VCO to the Radio Carrier Frequency.

The Low Port Baseband Frequency Word is added as modulation to the Sigma-Delta input.

45.3.2.5.1 Low Port Resolution

The table below shows the PLL Minimum Frequency Step Sizes for Modulation and Carrier Frequency tuning.

Note that the minimum step size for carrier frequency tuning is a divide by 512 of the modulation step sizes.

Table 45-14. PLL Minimum Frequency Step Size

Ref Clk Freq (MHz)	32	26
PLL Minimum Modulation Step (Hz)	244.14	198.36
PLL Minimum Carrier Frequency Step (Hz)	0.9537	0.7749

45.3.2.5.2 Low Port Sigma-Delta

The Integer, Numerator, and Denominator are presented at the input to the Sigma-Delta Modulator.

A 3-level quantized Sigma-Delta output signal {-1, 0, +1} is generated and combined with the Integer divide value to produce the Modulus Divide Ratio for the PLL Loop Divider. The resulting RF Frequency can be calculated using the formula below:

RF 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.

45.3.2.5.2.1 LPM SDM Dither

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 LPM 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.

This dither operation can also be overridden using the LPM_D_OVRD and LPM_D_CTRL bits.

45.3.2.5.3 Low Port Delay, Disable, Invert

The Low Port Modulator has an array of programmable delays available to allow the HPM modulation to be matched to the LPM modulation and the 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 LPM 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.

The Low Port Modulation of the Loop Divider can be disabled by setting the LPM_DISABLE bit. When the modulation is disabled the LPM SDM will continue to tune the VCO to the Radio Carrier Frequency.

45.3.2.6 Lock Detector

The PLL Lock Detector has three methods to measure the stability of the PLL and check that it is tuned to the selected Radio Frequency.

The real-time results of the PLL Lock Detect module can be read as register bits, and flags are set and can be cleared to provide software long-term access to any PLL unlock events.

The TSM can be configured to choose any or all of the PLL Lock Detect module methods to generate an Abort and avoid violating any FCC On-Air or other radio requirements. This abort option is also a power saving feature and avoids continuing a radio operation that will be unsuccessful.

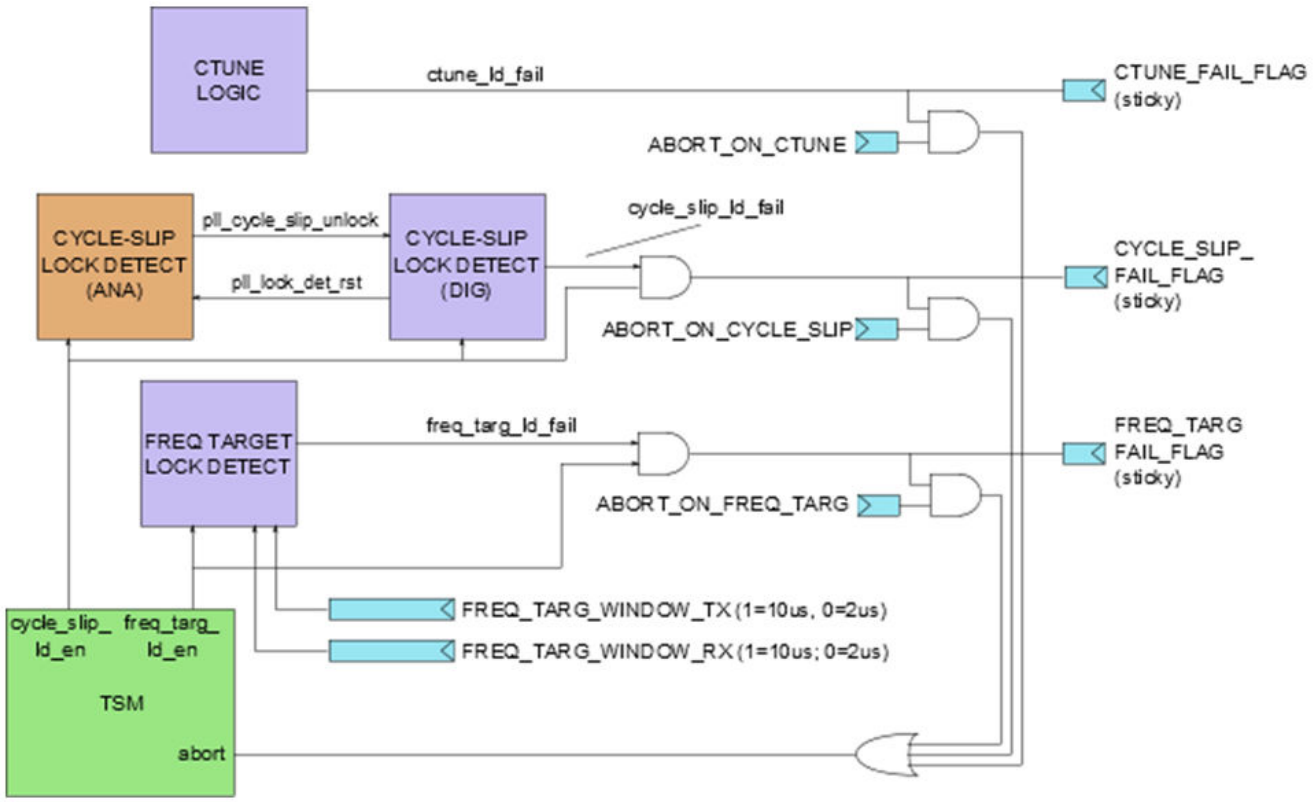


Figure 45-23. PLL Lock Detect Block Diagram

45.3.2.6.1 Coarse Tune Fail

During every Radio start-up sequence the PLL does a Coarse Tune Calibration to select a VCO setting that will allow the PLL to correctly tune and lock the VCO when the PLL/VCO loop is closed.

During the coarse tune calibration a count is maintained that indicates how close the coarse tune is to the targeted Radio Frequency, if this count is outside of the threshold selected by the Coarse Tune Lock Detect Fail Level register (CTUNE_LDF_LEV), then the Coarse Tune Failure Flag bit (CTFF) will be set and it can be used to abort the TSM sequence.

The CTFF bit is cleared by writing a 1 to it.

The CT_FAIL bit always shows the real time status of the latest Coarse Tune Calibration. If the absolute count difference was out of the range selected by CTUNE_LDF_LEV, then this bit will be set.

45.3.2.6.2 Fine Tune Fail

At the end of every Radio start-up sequence the PLL turns on the PLL Frequency Meter for a time that is requested by the Frequency Target Window time select register bits, separately selected for RX (FTW_RX) and TX (FTW_TX).

These bits select either a 4us or an 8us count period and if the frequency counter is outside of the threshold selected by the Frequency Target Fail Threshold registers, FTF_RX_THRSH and FTF_TX_THRSH, then the Frequency Target Fail Flag bit (FTFF) will be set and it can be used to abort the TSM sequence.

The FTFF bit is cleared by writing a 1 to it

The FT_FAIL bit always shows the real time status of the latest Fine Tune Frequency Measurement. If the absolute count difference was out of the range selected by Frequency Target Fail Threshold registers, then this bit will be set.

45.3.2.6.3 Cycle Slip Fail

During any radio sequence, the PLL can turn on an analog cycle slip detection circuit that measures the phase difference between the VCO and the reference clock. If the phase is not properly aligned the circuit will send a signal to the PLL Lock Detect digital circuit.

The PLL Lock Detect digital circuit evaluates the number and timing of the analog cycle slip events and can generate a Cycle Slip Fail Flag (CSFF).

There is a programmable state machine that controls the monitoring of Cycle Slip Events as shown in the figure below. The CS_FAIL bit always shows the real time status of the cycle slip state machine.

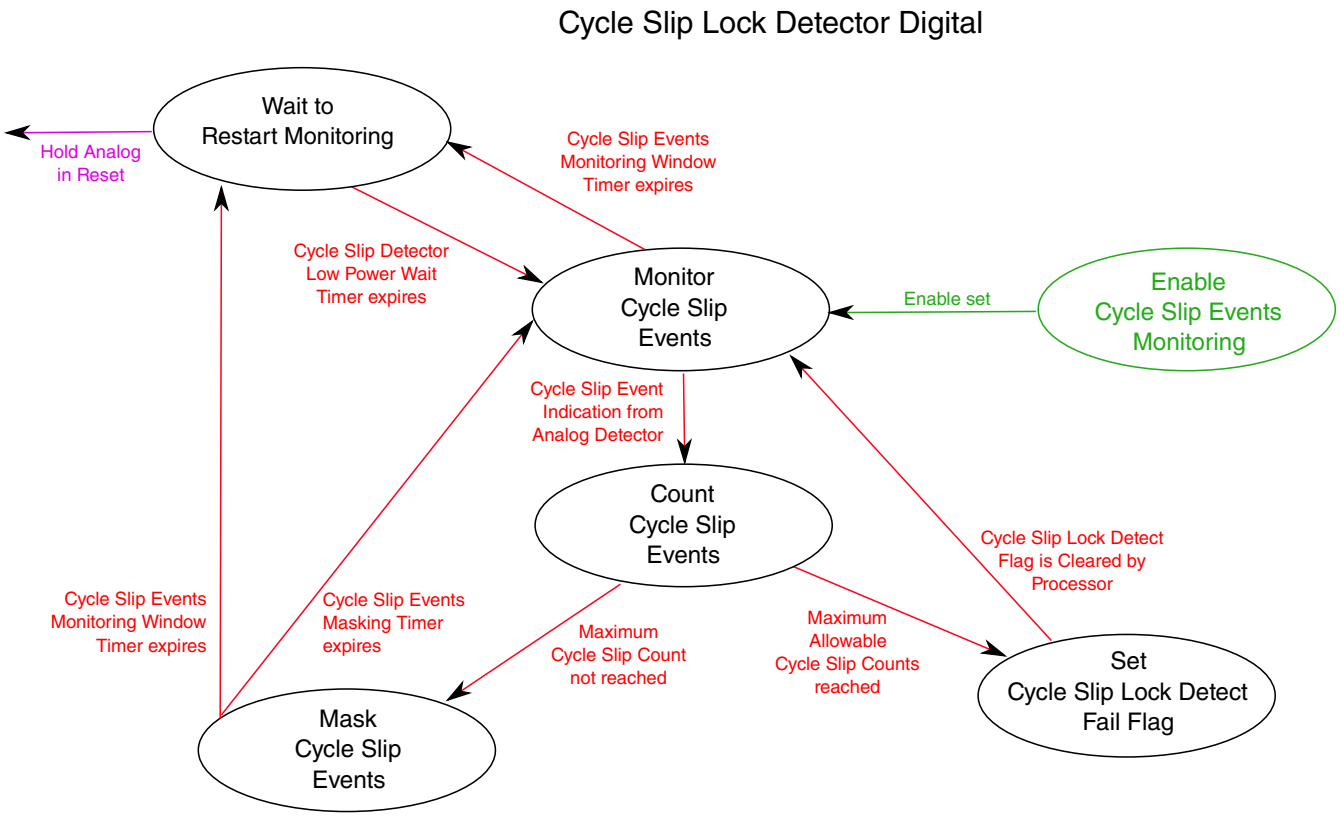


Figure 45-24. PLL Lock Detect State Machine

The following table below shows an example calculation of the time needed for the Cycle Slip Lock Detector to fail as a function of the number of cycle slip events and the VCO frequency error.

Crystal frequency	32	MHz							
Reference Time Period	31.25	ns							
				3	4	5	8	10	15
Frequency Error (kHz)	Loop Divider Time Period(ns)	Num cycles in 32MHz Time Period	Cycle-Slip Time (us)	Time for 3 cycle-slips (us)	Time for 4 cycle-slips (us)	Time for 5 cycle-slips (us)	Time for 8 cycle-slips (us)	Time for 10 cycle-slips (us)	Time for 15 cycle-slips (us)
-5000	37.04	5.400	**0.169**	0.506	1.519	4.556	13.669	41.006	123.019
-2000	33.33	15.000	**0.469**	1.406	4.219	12.656	37.969	113.906	341.719
-1000	32.26	31.000	**0.969**	2.906	8.719	26.156	78.469	235.406	706.219
-500	31.75	63.000	**1.969**	5.906	17.719	53.156	159.469	478.406	1435.219
-200	31.45	159.000	**4.969**	14.906	44.719	134.156	402.469	1207.406	3622.219
150	31.10	214.333	**6.698**	20.094	60.281	180.844	542.531	1627.594	4882.781
-100	31.35	319.000	**9.969**	29.906	89.719	269.156	807.469	2422.406	7267.219
0	31.25								
100	31.15	321.000	**10.031**	30.094	90.281	270.844	812.531	2437.594	7312.781
150	31.10	214.333	**6.698**	20.094	60.281	180.844	542.531	1627.594	4882.781
200	31.06	161.000	**5.031**	15.094	45.281	135.844	407.531	1222.594	3667.781
500	30.77	65.000	**2.031**	6.094	18.281	54.844	164.531	493.594	1480.781
1000	30.30	33.000	**1.031**	3.094	9.281	27.844	83.531	250.594	751.781
2000	29.41	17.000	**0.531**	1.594	4.781	14.344	43.031	129.094	387.281
5000	27.03	7.400	**0.231**	0.694	2.081	6.244	18.731	56.194	168.581
10000	23.81	4.200	**0.131**	0.394	1.181	3.544	10.631	31.894	95.681
20000	19.23	2.600	**0.081**	0.244	0.731	2.194	6.581	19.744	59.231
50000	12.20	1.640	**0.051**	0.154	0.461	1.384	4.151	12.454	37.361

Figure 45-25. Cycle Slip Monitoring Window as a function of frequency error and number of cycle slip events

A PLL Cycle Slip Fail example is shown in the figure below.

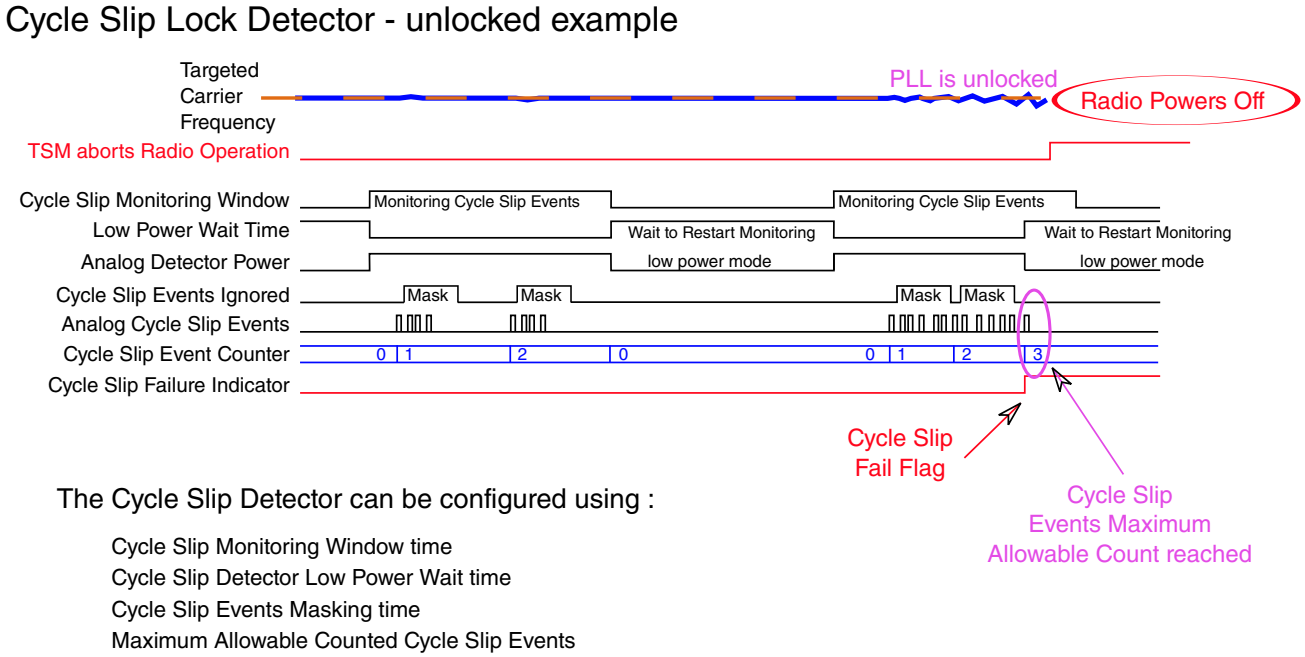


Figure 45-26. PLL Unlock Example

45.3.2.6.3.1 Cycle Slip Events Monitor Window

The Cycle Slip Events Monitoring Window Timer selects the time in microseconds that the Cycle Slip Lock Detector will monitor Cycle Slip Events.

If the Cycle Slip Events Monitoring Window Timer expires before the Cycle Slip Events Count is reached, then the Events Counter will be reset and the Events Counting sequence can restart after the Cycle Slip Detector Low Power Wait Time if the Cycle Slip Recycle bit is set.

The following selections are to be supported for the Cycle Slip Events Monitoring Window Time:

Table 45-15. Cycle Slip Events Monitoring Window Times

CS_FW[2:0], Cycle Slip Monitor Time	Monitor Time (us)
0	8
1	16
2	24
3	32
4	64

Table continues on the next page...

Table 45-15. Cycle Slip Events Monitoring Window Times (continued)

5	96
6	128
7	256

45.3.2.6.3.2 Cycle Slip Detector Low Power Wait Timer

The Cycle Slip Detector Low Power Wait Timer is a method for saving power by turning off the analog cycle slip detector circuit for a programmable period of time. If the Cycle Slip Recycle bit is set when the Cycle Slip Events Monitoring Window Timer expires, then the Events Counter will be reset and the Events Counting sequence will restart after the Cycle Slip Detector Low Power Wait Time expires.

The following selections are to be supported for the Low Power Wait Time:

Table 45-16. Cycle Slip Detector Low Power Wait Times

CS_WT[2:0], Cycle Slip Wait Time	Wait Time (us)
0	128
1	256
2	384
3	512
4	640
5	768
6	896
7	1024

45.3.2.6.3.3 Cycle Slip Events Masking Timer

The Cycle Slip Events Masking Timer selects the number of nanoseconds that the Cycle Slip Lock Detector will wait before counting the next Cycle Slip Event. This allows time for the PLL to try to regain lock on the VCO.

The following selections are to be supported for the Cycle Slip Events Masking time after an analog cycle slip event:

Table 45-17. Cycle Slip Events Masking Times

CS_FT[4:0], Cycle Slips Masking Time	Wait Time (ns), 26 MHz	Wait Time (ns), 32 MHz
0	38	31

Table continues on the next page...

Table 45-17. Cycle Slip Events Masking Times (continued)

1	77	63
2	115	94
3	154	125
4	192	156
5	231	188
6	269	219
7	308	250
8	346	281
9	385	313
10	423	344
11	462	375
12	500	406
13	538	438
14	577	469
15	615	500
16	654	531
17	692	563
18	731	594
19	769	625
20	808	656
21	846	688
22	885	719
23	923	750
24	962	781
25	1000	813
26	1038	844
27	1077	875
28	1115	906
29	1154	938
30	1192	969
31	1231	1000

45.3.2.6.3.4 Cycle Slip Events Count

The Cycle Slip Events Count (CS_FCNT) determines the number of Cycle Slip Events that will cause a Cycle Slip Fail Flag to be set.

The range of Cycle Slip Events Counted that will cause a Fail Flag shall be from 1 to 15.

45.3.2.7 Memory Map and Register Definition

The PLL Digital memory map and detailed descriptions of all its registers are as follows.

45.3.2.7.1 XCVR_PLL_DIG register descriptions

45.3.2.7.1.1 XCVR_PLL_DIG_ADDR Memory map

XCVR_PLL_DIG base address: 4005_C224h

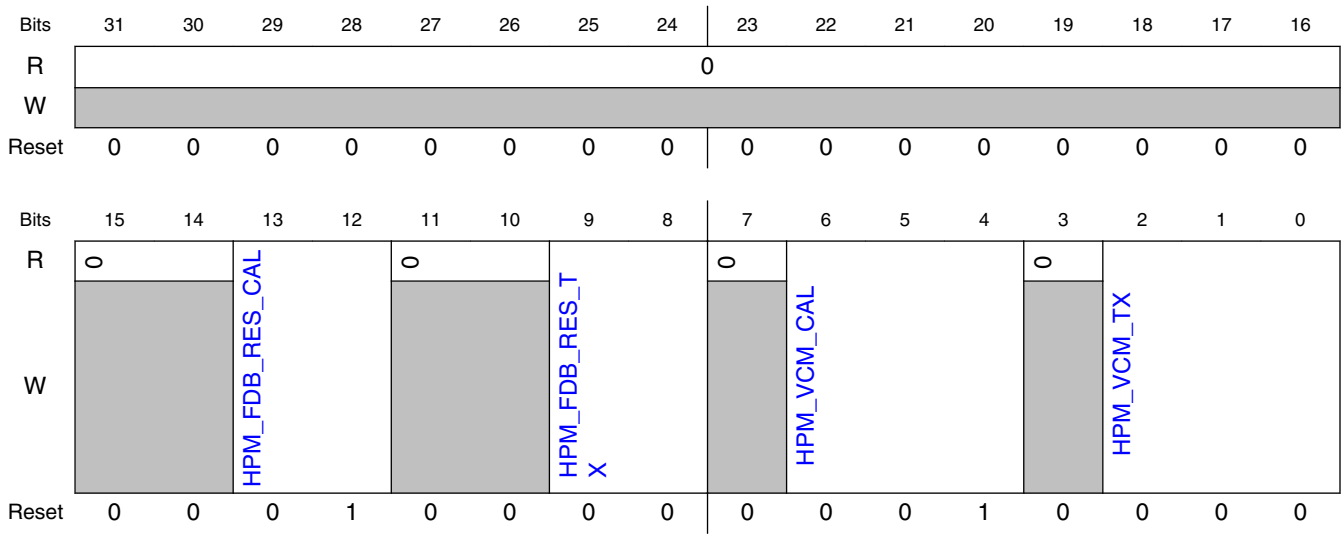
Offset	Register	Width (In bits)	Access	Reset value
0h	PLL HPM Analog Bump Control (HPM_BUMP)	32	RW	0000_1010h
4h	PLL Modulation Control (MOD_CTRL)	32	RW	0000_0000h
8h	PLL Channel Mapping (CHAN_MAP)	32	RW	0000_0200h
Ch	PLL Lock Detect Control (LOCK_DETECT)	32	RW	0060_6800h
10h	PLL High Port Modulator Control (HPM_CTRL)	32	RW	9084_0000h
20h	PLL High Port Sigma Delta Results (HPM_SDM_RES)	32	RW	0100_0000h
24h	PLL Low Port Modulator Control (LPM_CTRL)	32	RW	0202_0000h
28h	PLL Low Port Sigma Delta Control 1 (LPM_SDM_CTRL1)	32	RW	0026_0026h
2Ch	PLL Low Port Sigma Delta Control 2 (LPM_SDM_CTRL2)	32	RW	0008_0000h
30h	PLL Low Port Sigma Delta Control 3 (LPM_SDM_CTRL3)	32	RW	0010_0000h
34h	PLL Low Port Sigma Delta Result 1 (LPM_SDM_RES1)	32	RO	0000_0000h
38h	PLL Low Port Sigma Delta Result 2 (LPM_SDM_RES2)	32	RO	0400_0000h
3Ch	PLL Delay Matching (DELAY_MATCH)	32	RW	0000_0004h
40h	PLL Coarse Tune Control (CTUNE_CTRL)	32	RW	0000_0000h
54h	PLL Coarse Tune Results (CTUNE_RES)	32	RO	0962_0040h

45.3.2.7.1.2 PLL HPM Analog Bump Control (HPM_BUMP)

45.3.2.7.1.2.1 Offset

Register	Offset
HPM_BUMP	0h

45.3.2.7.1.2.2 Diagram



45.3.2.7.1.2.3 Fields

Field	Function
31-14 —	Reserved
13-12 HPM_FDB_RES_CAL	rfctrl_tx_dac_bump_fdb_res[1:0] during Calibration This is the value applied to the rfctrl_tx_dac_bump_fdb_res[1:0] port during High Port Calibration. This register sets the HPM DAC feedback resistor to increase the modulation gain during calibration. 00b - 29 kohms 01b - 58 kohms(gain of 2) 10b - 13 kohms 11b - 23.7 kohms
11-10 —	Reserved
9-8 HPM_FDB_RES_TX	rfctrl_tx_dac_bump_fdb_res[1:0] during Transmission This is the value applied to the rfctrl_tx_dac_bump_fdb_res[1:0] port during Radio Transmissions. This register sets the HPM DAC feedback resistor during transmissions. 00b - 29 kohms 01b - 58 kohms(gain of 2) 10b - 13 kohms 11b - 23.7 kohms
7 —	Reserved
6-4 HPM_VCM_CAL	rfctrl_tx_dac_bump_vcm[2:0] during Calibration This is the value applied to the rfctrl_tx_dac_bump_vcm[2:0] port during High Port Calibration. This register sets the HPM DAC Op-Amp reference voltage during calibration.

Table continues on the next page...

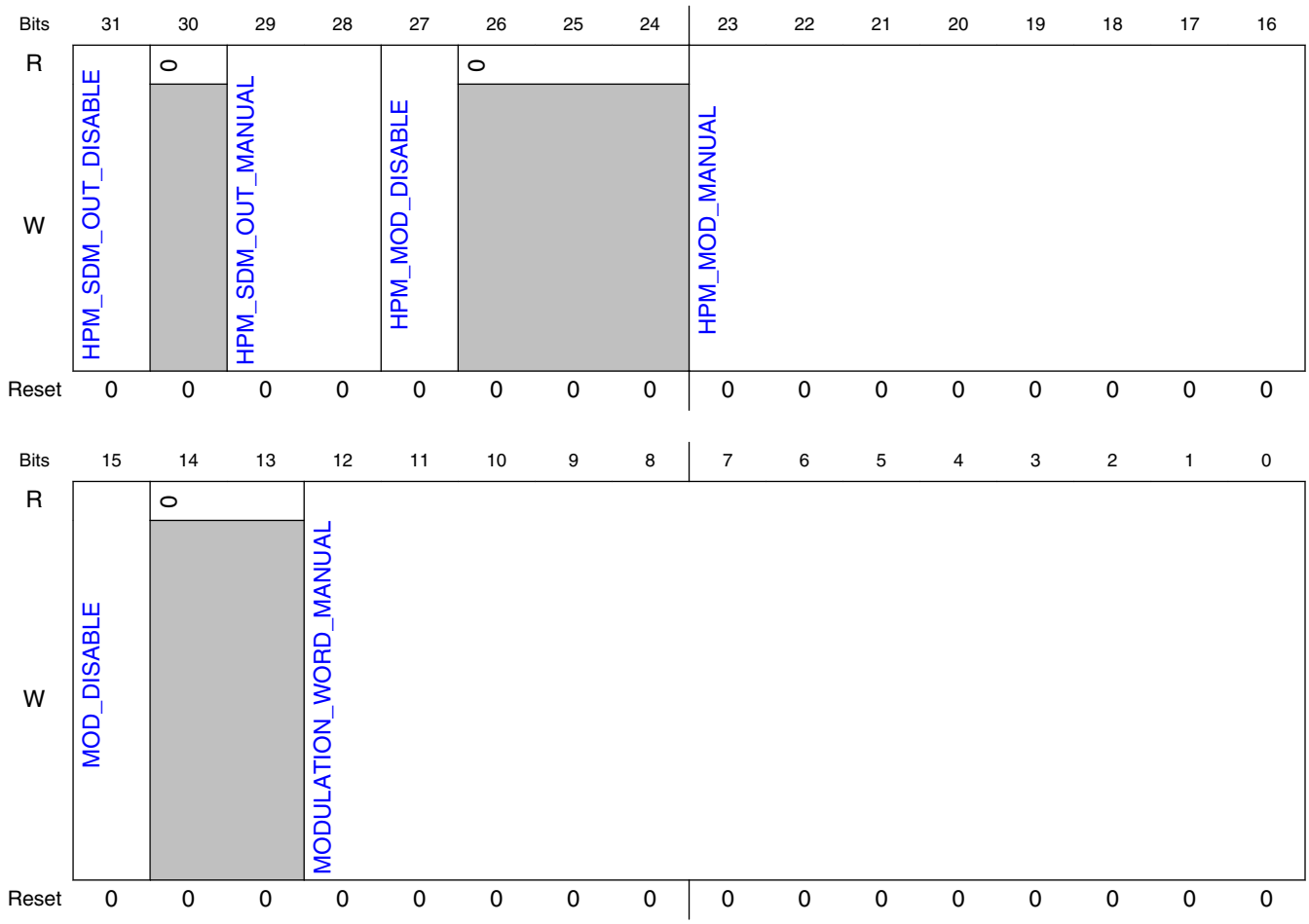
Field	Function
	000b - 432 mV 001b - 328 mV 010b - 456 mV 011b - 473 mV 100b - 488 mV 101b - 408 mV 110b - 392 mV 111b - 376 mV
3 —	Reserved
2-0 HPM_VCM_TX	rfctrl_tx_dac_bump_vcm[2:0] during Transmission This is the value applied to the rfctrl_tx_dac_bump_vcm[2:0] port during Radio Transmissions. This register sets the HPM DAC Op-Amp reference voltage during transmissions. 000b - 432 mV 001b - 328 mV 010b - 456 mV 011b - 473 mV 100b - 488 mV 101b - 408 mV 110b - 392 mV 111b - 376 mV

45.3.2.7.1.3 PLL Modulation Control (MOD_CTRL)

45.3.2.7.1.3.1 Offset

Register	Offset
MOD_CTRL	4h

45.3.2.7.1.3.2 Diagram



45.3.2.7.1.3.3 Fields

Field	Function
31 HPM_SDM_OUT_DISABLE	Disable HPM SDM out If this bit is set, the High Port Sigma Delta Modulator output is disabled, and the High Port Fractional value applied to the VCO comes from the HPM_SDM_OUT_MANUAL register.
30 —	Reserved
29-28 HPM_SDM_OUT_MANUAL	Manual HPM SDM out If HPM_SDM_OUT_DISABLE is set, this register is the Fractional value that is applied to the VCO High Port.
27 HPM_MOD_DISABLE	Disable HPM Modulation If this bit is set, the High Port Modulation is disabled, and the High Port Modulation value applied to the VCO comes from the HPM_MOD_MANUAL register.
26-24 —	Reserved

Table continues on the next page...

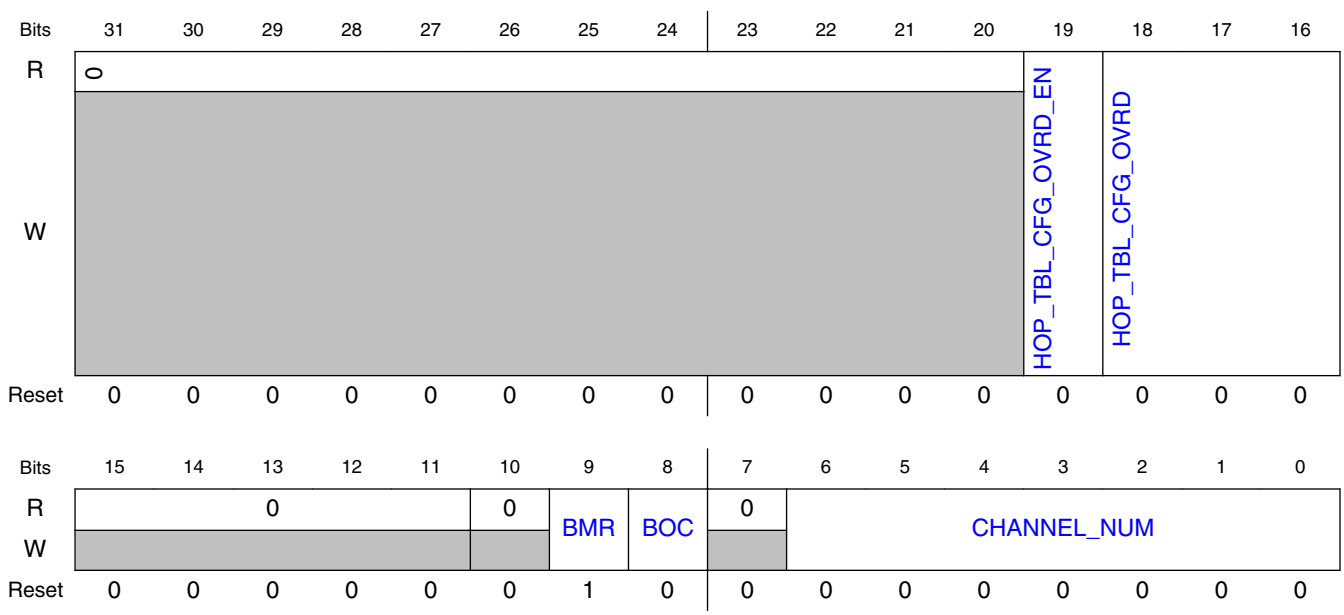
Field	Function
23-16 HPM_MOD_MANUAL	Manual HPM Modulation If HPM_MOD_DISABLE is set, this register is the modulation value that is applied to the VCO High Port.
15 MOD_DISABLE	Disable Modulation Word If this bit is set, any modulation from the TX Digital is disabled and the source of the Baseband Frequency Word is the MODULATION_WORD_MANUAL register.
14-13 —	Reserved
12-0 MODULATION_WORD_MANUAL	Manual Modulation Word If MOD_DISABLE is set, the signed 12 bit value that is represented by this register is the Baseband Frequency Word.

45.3.2.7.1.4 PLL Channel Mapping (CHAN_MAP)

45.3.2.7.1.4.1 Offset

Register	Offset
CHAN_MAP	8h

45.3.2.7.1.4.2 Diagram



45.3.2.7.1.4.3 Fields

Field	Function
31-20 —	Reserved
19 HOP_TBL_CFG_OVRD_EN	Hop Table Configuration Override Enable If this bit is set then the HOP_TBL_CFG_OVRD and DFT_PATTERN Registers will be used to control the Coarse Tune, Channel Number, and Hop Table for the PLL Carrier Frequency Selection.
18-16 HOP_TBL_CFG_OVRD	Hop Table Configuration Override If the HOP_TBL_CFG_OVRD_EN bit is set, then the DFT_PATTERN Register will be used to control the Hop Table for the PLL Carrier Frequency Selection. 010b - DFT_PATTERN[15:7] is signed offset to DFT_PATTERN[6:0] mapped channel number 011b - DFT_PATTERN[15:1] is signed Numerator, DFT_PATTERN[0] is integer selection
15-11 —	Reserved
10 —	Reserved
9 BMR	BLE MBAN Channel Remap This bit controls the mapping of BLE channel 39 in Radio Protocol 2, BLE overlap MBAN mode. 0b - BLE channel 39 is mapped to BLE channel 39, 2.480 GHz 1b - 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. 0b - BLE channel number comes from the BLE Link Layer 1b - BLE channel number comes from the CHANNEL_NUM register (BLE protocols 0 and 2)
7 —	Reserved
6-0 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 protocols 1, 6 and 7. The Radio Channel Frequency can also be selected by setting the SDM_MAP_DISABLE bit in the PLL_LPM_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).

Radio Protocols Supported, Channel Frequency in MHz					
BLE		MBAN		BLE/MBAN	
0		1		2	
Chan	MHz	Chan	MHz	Chan	MHz
0	2,402	0	2,360	0	2,402

Field	Function					
	1	2,404	1	2,361	1	2,404
	2	2,406	2	2,362	2	2,406
	3	2,408	3	2,363	3	2,408
	4	2,410	4	2,364	4	2,410
	5	2,412	5	2,365	5	2,412
	6	2,414	6	2,366	6	2,414
	7	2,416	7	2,367	7	2,416
	8	2,418	8	2,368	8	2,418
	9	2,420	9	2,369	9	2,420
	10	2,422	10	2,370	10	2,422
	11	2,424	11	2,371	11	2,424
	12	2,426	12	2,372	12	2,426
	13	2,428	13	2,373	13	2,428
	14	2,430	14	2,374	14	2,430
	15	2,432	15	2,375	15	2,432
	16	2,434	16	2,376	16	2,434
	17	2,436	17	2,377	17	2,436
	18	2,438	18	2,378	18	2,438
	19	2,440	19	2,379	19	2,440
	20	2,442	20	2,380	20	2,442
	21	2,444	21	2,381	21	2,444
	22	2,446	22	2,382	22	2,446
	23	2,448	23	2,383	23	2,448
	24	2,450	24	2,384	24	2,450
	25	2,452	25	2,385	25	2,452
	26	2,454	26	2,386	26	2,454
	27	2,456	27	2,387	27	2,456
	28	2,458	28	2,388	28	2,458
	29	2,460	29	2,389	29	2,460
	30	2,462	30	2,390	30	2,390
	31	2,464	31	2,391	31	2,391
	32	2,466	32	2,392	32	2,392
	33	2,468	33	2,393	33	2,393
	34	2,470	34	2,394	34	2,394
	35	2,472	35	2,395	35	2,395
	36	2,474	36	2,396	36	2,396
	37	2,476	37	2,397	37	2,397
	38	2,478	38	2,398	38	2,398
	39	2,480	39	2,399	39*	2,480

* The BLE MBAN Channel Remap bit, BMR, controls the frequency mapping in this case.

Field	Function

Radio Protocols Supported, Channel Frequency in MHz					
DFT, Generic					
6,7,8,9					
Channel	MHz	Channel	MHz	Channel	MHz
0	2,360	43	2,403	86	2,446
1	2,361	44	2,404	87	2,447
2	2,362	45	2,405	88	2,448
3	2,363	46	2,406	89	2,449
4	2,364	47	2,407	90	2,450
5	2,365	48	2,408	91	2,451
6	2,366	49	2,409	92	2,452
7	2,367	50	2,410	93	2,453
8	2,368	51	2,411	94	2,454
9	2,369	52	2,412	95	2,455
10	2,370	53	2,413	96	2,456
11	2,371	54	2,414	97	2,457
12	2,372	55	2,415	98	2,458
13	2,373	56	2,416	99	2,459
14	2,374	57	2,417	100	2,460
15	2,375	58	2,418	101	2,461
16	2,376	59	2,419	102	2,462
17	2,377	60	2,420	103	2,463
18	2,378	61	2,421	104	2,464
19	2,379	62	2,422	105	2,465
20	2,380	63	2,423	106	2,466
21	2,381	64	2,424	107	2,467
22	2,382	65	2,425	108	2,468
23	2,383	66	2,426	109	2,469
24	2,384	67	2,427	110	2,470
25	2,385	68	2,428	111	2,471
26	2,386	69	2,429	112	2,472
27	2,387	70	2,430	113	2,473
28	2,388	71	2,431	114	2,474
29	2,389	72	2,432	115	2,475
30	2,390	73	2,433	116	2,476
31	2,391	74	2,434	117	2,477
32	2,392	75	2,435	118	2,478
33	2,393	76	2,436	119	2,479
34	2,394	77	2,437	120	2,480
35	2,395	78	2,438	121	2,481

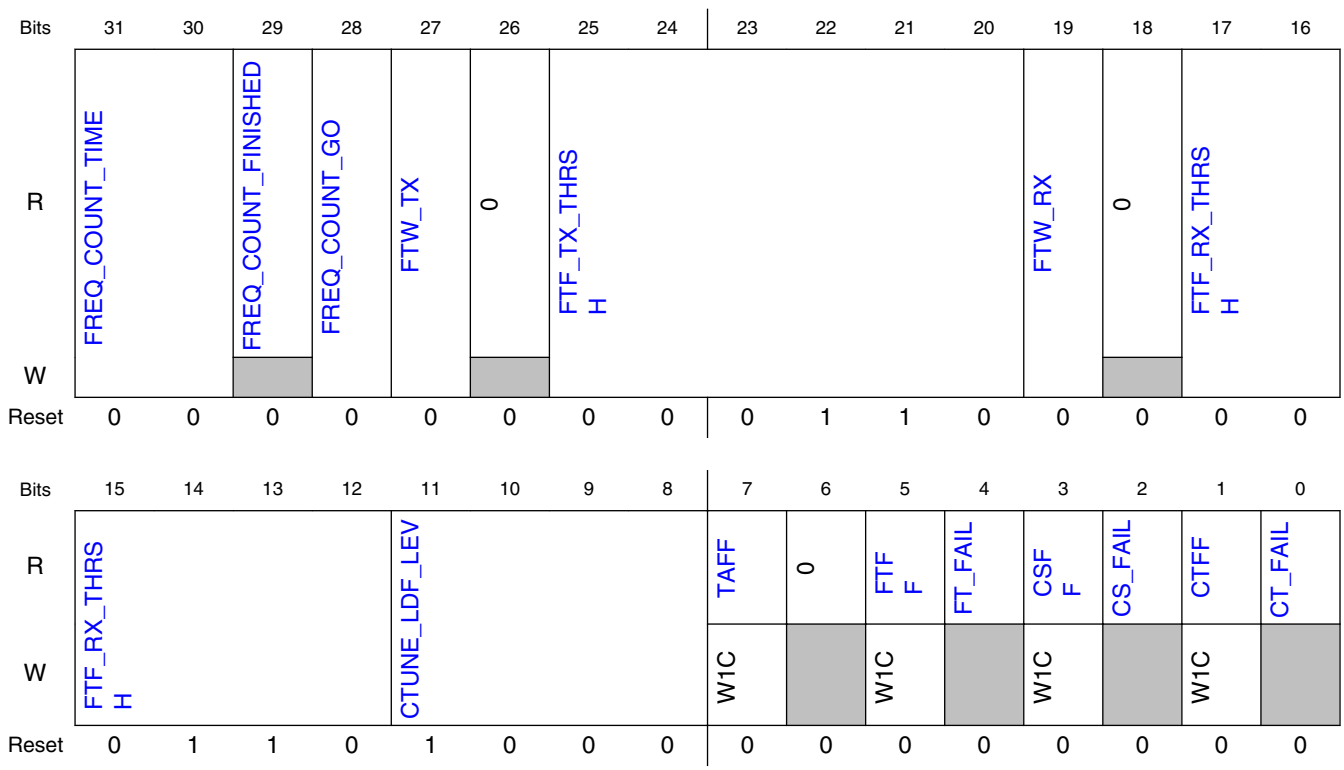
Field	Function					
	36	2,396	79	2,439	122	2,482
	37	2,397	80	2,440	123	2,483
	38	2,398	81	2,441	124	2,484
	39	2,399	82	2,442	125	2,485
	40	2,400	83	2,443	126	2,486
	41	2,401	84	2,444	127	2,487
	42	2,402	85	2,445		

45.3.2.7.1.5 PLL Lock Detect Control (LOCK_DETECT)

45.3.2.7.1.5.1 Offset

Register	Offset
LOCK_DETECT	Ch

45.3.2.7.1.5.2 Diagram



45.3.2.7.1.5.3 Fields

Field	Function
31-30 FREQ_COUNT_TIME	Frequency Meter Count Time This is the length of time that the Frequency Meter will count VCO clocks. 00b - 800 us 01b - 25 us 10b - 50 us 11b - 100 us
29 FREQ_COUNT_FINISHED	Frequency Meter has finished the Count Time This bit is set when the FREQ_COUNT_TIME value is reached, it is cleared when FREQ_COUNT_GO is cleared.
28 FREQ_COUNT_GO	Start the Frequency Meter The Frequency Meter starts when this bit is set and runs until the FREQ_COUNT_TIME value is reached. The bit should not be cleared until after the counting is finished. After the Counting Time finishes, the FREQ_COUNT_FINISHED bit will read as a one. Then the measured frequency can be calculated by reading the frequency counts in the DFT_FREQ_COUNTER register (in the XCVR_ANALOG register space) and dividing by the FREQ_COUNT_TIME. Note that the counting is done at the VCO frequency which is 2X the Carrier Frequency
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. 0b - 4 us 1b - 8 us
26 —	Reserved
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. 0b - 4 us 1b - 8 us
18 —	Reserved
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	TSM Abort Failure Flag

Table continues on the next page...

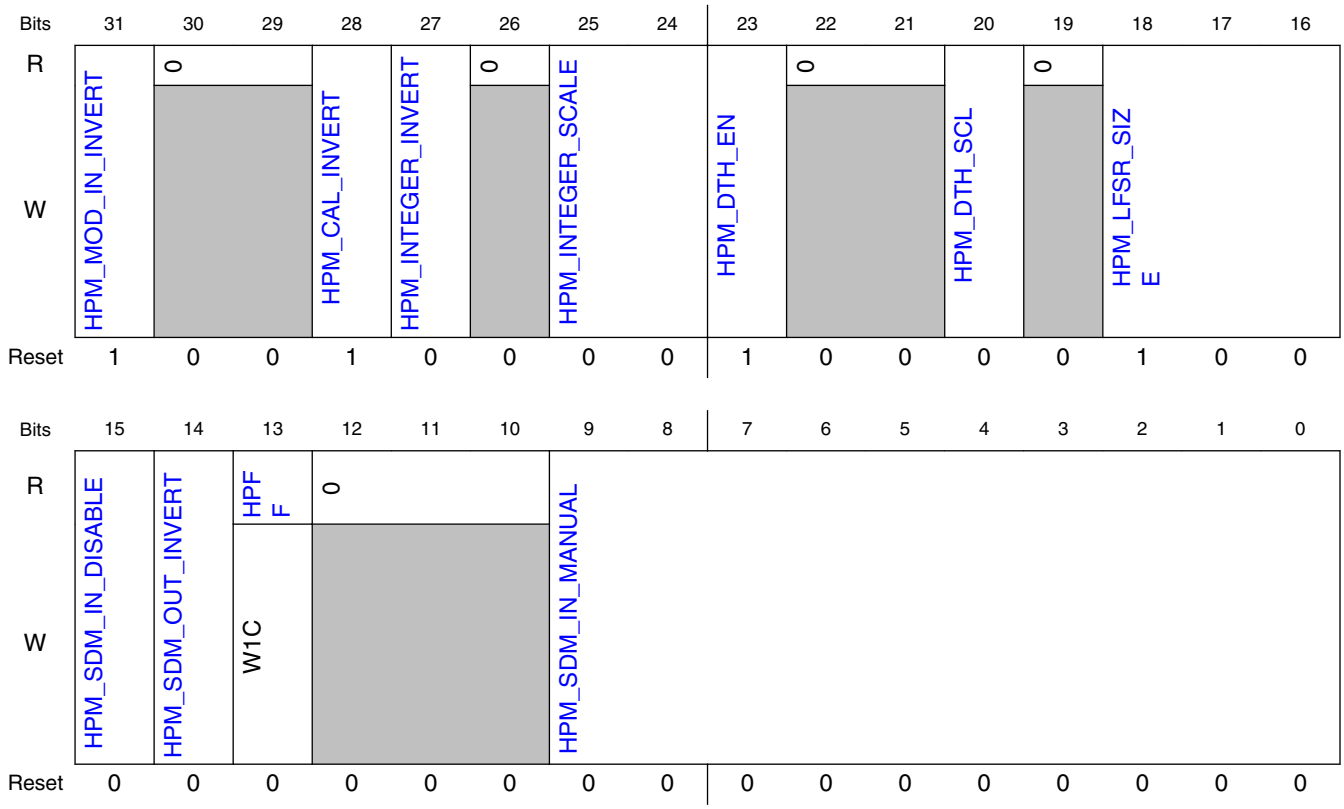
Field	Function
TAFF	This bit is set if the TSM Sequence Aborts, and this bit is cleared by writing a 1 to it.
6 —	Reserved
5 FTFF	Frequency Target Failure Flag This bit is set when FT_FAIL is first set, and this bit is cleared by writing a 1 to it.
4 FT_FAIL	Real time status of Frequency Target Failure 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.
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.

45.3.2.7.1.6 PLL High Port Modulator Control (HPM_CTRL)

45.3.2.7.1.6.1 Offset

Register	Offset
HPM_CTRL	10h

45.3.2.7.1.6.2 Diagram



45.3.2.7.1.6.3 Fields

Field	Function
31 HPM_MOD_IN_INVERT	Invert High Port Modulation If this bit is set then the High Port Modulation Word is inverted before it is multiplied and split into Integer and Fractional values.
30-29 —	Reserved
28 HPM_CAL_INVERT	Invert High Port Modulator Calibration If this bit is set then the order of the High Port Calibration is reversed in order to get a positive count difference between the Maximum and Minimum settings of the HPM DAC.
27 HPM_INTEGER_INVERT	Invert High Port Modulation Integer If this bit is set then the High Port Modulation Integer value, after the multiply and split into Integer and Fractional values, will be inverted before it is applied to the VCO High Port DAC Array.
26 —	Reserved
25-24 HPM_INTEGER_SCALE	High Port Modulation Integer Scale This register controls the scaling of the High Port Modulation Integer Value applied to the VCO High Port DAC Array. 00b - No Scaling

Table continues on the next page...

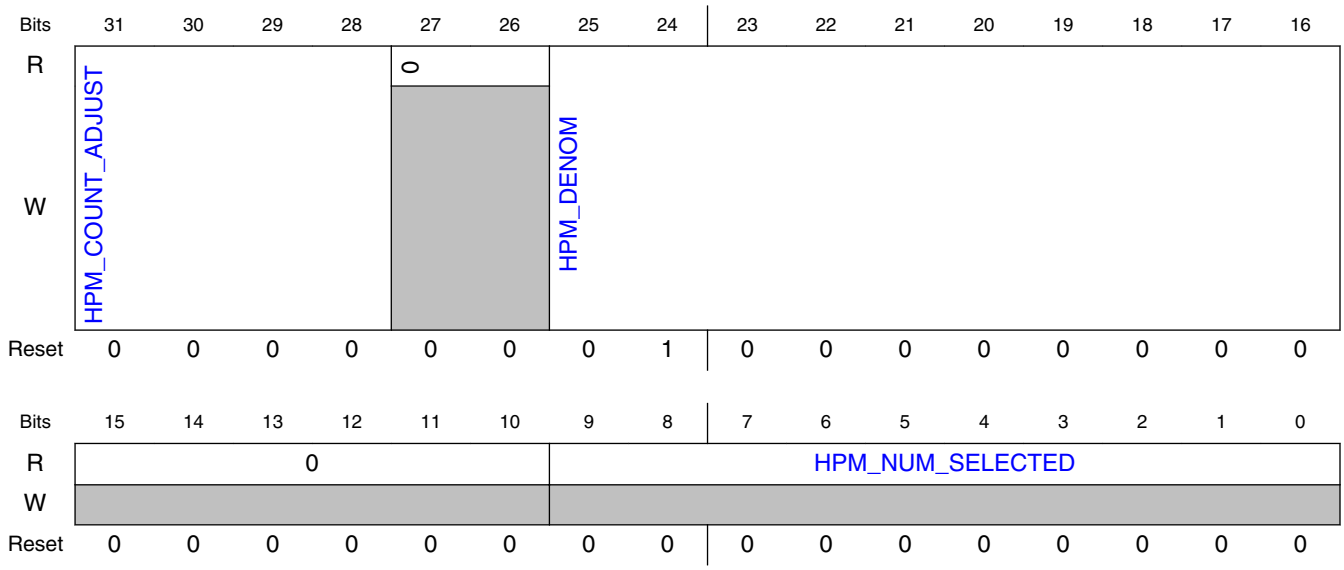
Field	Function
	01b - Multiply by 2 10b - Divide by 2 11b - Reserved
23 HPM_DTH_EN	Dither Enable for HPM LFSR If this bit is set, the High Port Fraction will be Dithered by the High Port LFSR before it is applied to the High Port SDM.
22-21 —	Reserved
20 HPM_DTH_SCL	HPM Dither Scale If this bit is set, the LFSR dithering of the High Port Fraction will be multiplied by 2.
19 —	Reserved
18-16 HPM_LFSR_SIZE	HPM LFSR Length This register selects the length of the HPM LFSR and the associated LFSR Tap Mask 000b - LFSR 9, tap mask 100010000 001b - LFSR 10, tap mask 1001000000 010b - LFSR 11, tap mask 11101000000 011b - LFSR 13, tap mask 1101100000000 100b - LFSR 15, tap mask 111010000000000 101b - LFSR 17, tap mask 11110000000000000 110b - Reserved 111b - Reserved
15 HPM_SDM_IN_DISABLE	Disable HPM SDM Input If this bit is set, the Fractional portion of the High Port Modulation to the High Port SDM is disabled, and the High Port SDM input comes from the HPM_SDM_IN_MANUAL register.
14 HPM_SDM_OUT_INVERT	Invert HPM SDM Output If this bit is set the High Port SDM result will be Inverted before it is applied to the VCO High Port DAC Array.
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
9-0 HPM_SDM_IN_MANUAL	Manual High Port SDM Fractional value If HPM_SDM_IN_DISABLE is set, this register is the value that is applied as the Fractional value to the input of the High Port SDM.

45.3.2.7.1.7 PLL High Port Sigma Delta Results (HPM_SDM_RES)

45.3.2.7.1.7.1 Offset

Register	Offset
HPM_SDM_RES	20h

45.3.2.7.1.7.2 Diagram



45.3.2.7.1.7.3 Fields

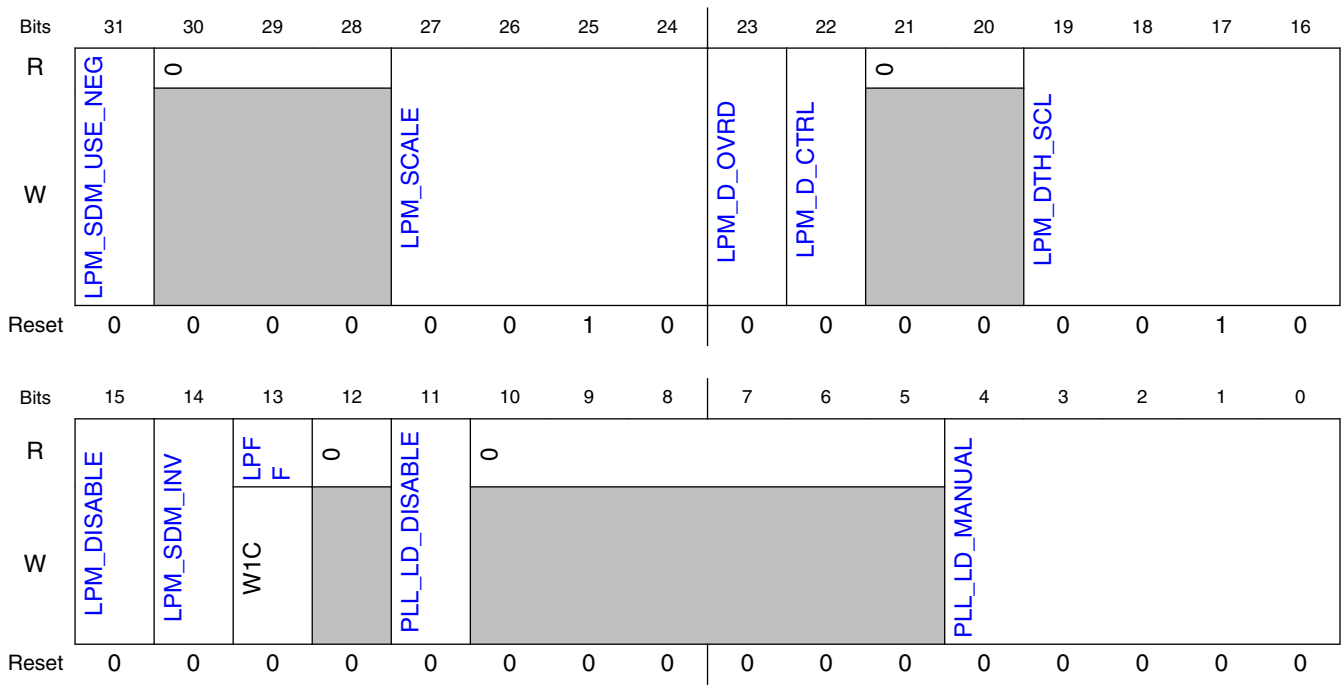
Field	Function
31-28 HPM_COUNT_ADJUST	HPM_COUNT_ADJUST This register represents a signed three bit value that is used to adjust the High Port Calibration Frequency Count Difference. The range of adjustment is -8 to +7, applied to the difference between Count 1 and Count 2.
27-26 —	Reserved
25-16 HPM_DENOM	High Port Modulator SDM 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
15-10 —	Reserved
9-0 HPM_NUM_SELECTED	High Port Modulator SDM Numerator 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

45.3.2.7.1.8 PLL Low Port Modulator Control (LPM_CTRL)

45.3.2.7.1.8.1 Offset

Register	Offset
LPM_CTRL	24h

45.3.2.7.1.8.2 Diagram



45.3.2.7.1.8.3 Fields

Field	Function
31 LPM_SDM_USE_NEG	Use the Negedge of the Sigma Delta clock If this bit is set then the negative edge of the Sigma Delta clock is used to launch the Low Port Modulation word to the VCO Loop Divider.
30-28 —	Reserved
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. 0000b - No Scaling 0001b - Multiply by 2 0010b - Multiply by 4 0011b - Multiply by 8 0100b - Multiply by 16 0101b - Multiply by 32

Table continues on the next page...

Field	Function
	0110b - Multiply by 64 0111b - Multiply by 128 1000b - Multiply by 256, this is the intended setting for normal operation. 1001b - Multiply by 512 1010b - Multiply by 1024 1011b - Multiply by 2048 1100b - Reserved 1101b - Reserved 1110b - Reserved 1111b - 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 in Radio Transmit mode 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
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. 0000b - Reserved 0001b - Reserved 0010b - Reserved 0011b - Reserved 0100b - Reserved 0101b - -128 to 96 0110b - -256 to 192 0111b - -512 to 384 1000b - -1024 to 768, this is the intended setting for normal operation. 1001b - -2048 to 1536 1010b - -4096 to 3072 1011b - -8192 to 6144 1100b - Reserved 1101b - Reserved 1110b - Reserved 1111b - Reserved
15 LPM_DISABLE	Disable LPM SDM This bit controls the Modulation of the Low Port Sigma Delta. If this bit is set, the Low Port Sigma Delta Modulator 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. No Modulation or Dithering will be added to the steady frequency result.
14 LPM_SDM_INV	Invert LPM SDM 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.

Table continues on the next page...

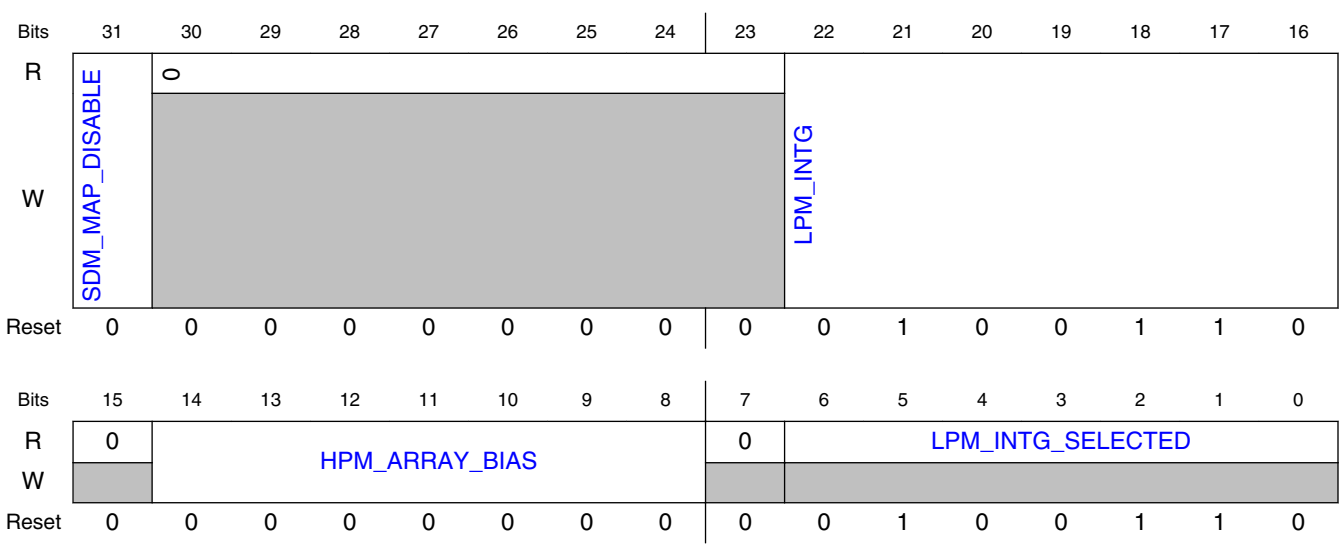
Field	Function
13 LPFF	LPM SDM Invalid Flag This bit is set if the Low Port Sigma Delta Modulator output is invalid due to an error in the fraction applied, and this bit is cleared by writing a 1 to it.
12 —	Reserved
11 PLL_LD_DISABLE	Disable PLL Loop Divider If this bit is set, the Low Port Sigma Delta Modulator output is disabled, and the PLL Loop Divider value applied to the PLL comes from the PLL_LD_MANUAL register.
10-5 —	Reserved
4-0 PLL_LD_MANUAL	Manual PLL Loop Divider value If PLL_LD_DISABLE is set, this register is the value that is applied to the PLL Loop Divider.

45.3.2.7.1.9 PLL Low Port Sigma Delta Control 1 (LPM_SDM_CTRL1)

45.3.2.7.1.9.1 Offset

Register	Offset
LPM_SDM_CTRL1	28h

45.3.2.7.1.9.2 Diagram



45.3.2.7.1.9.3 Fields

Field	Function
31 SDM_MAP_DISABLE	Disable SDM Mapping 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
22-16 LPM_INTG	Manual Low Port Modulation Integer Value If SDM_MAP_DISABLE 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 —	Reserved
14-8 HPM_ARRAY_BIAS	Bias value for High Port DAC Array Midpoint The value of this register represents a signed six bit value that can be used to adjust the midpoint of the High Port Modulator DAC. The range of adjustment is -64 to +63, and the adjustment is made to the High Port Modulation Word before the multiply, and before the split into Integer and Fractional values. The range of adjustment in Hz is approximately +/- 62.5 kHz
7 —	Reserved
6-0 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.

45.3.2.7.1.10 PLL Low Port Sigma Delta Control 2 (LPM_SDM_CTRL2)

45.3.2.7.1.10.1 Offset

Register	Offset
LPM_SDM_CTRL2	2Ch

45.3.2.7.1.10.2 Diagram

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

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

45.3.2.7.1.10.3 Fields

Field	Function
31-28 —	Reserved
27-0 LPM_NUM	Low Port Modulation Numerator If SDM_MAP_DISABLE 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

45.3.2.7.1.11 PLL Low Port Sigma Delta Control 3 (LPM_SDM_CTRL3)

45.3.2.7.1.11.1 Offset

Register	Offset
LPM_SDM_CTRL3	30h

45.3.2.7.1.11.2 Diagram

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

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

45.3.2.7.1.11.3 Fields

Field	Function
31-28 —	Reserved
27-0 LPM_DENOM	Low Port Modulation Denominator If SDM_MAP_DISABLE 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

45.3.2.7.1.12 PLL Low Port Sigma Delta Result 1 (LPM_SDM_RES1)

45.3.2.7.1.12.1 Offset

Register	Offset
LPM_SDM_RES1	34h

45.3.2.7.1.12.2 Diagram

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

45.3.2.7.1.12.3 Fields

Field	Function
31-28 —	Reserved
27-0 LPM_NUM_SELECTED	Low Port Modulation Numerator Applied This is the value that is currently being applied to the Low Port Sigma Delta Modulator, for valid Sigma Delta operation the resulting fraction NUM/DENOM must be in the range -0.6 to +0.6

45.3.2.7.1.13 PLL Low Port Sigma Delta Result 2 (LPM_SDM_RES2)

45.3.2.7.1.13.1 Offset

Register	Offset
LPM_SDM_RES2	38h

45.3.2.7.1.13.2 Diagram

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

45.3.2.7.1.13.3 Fields

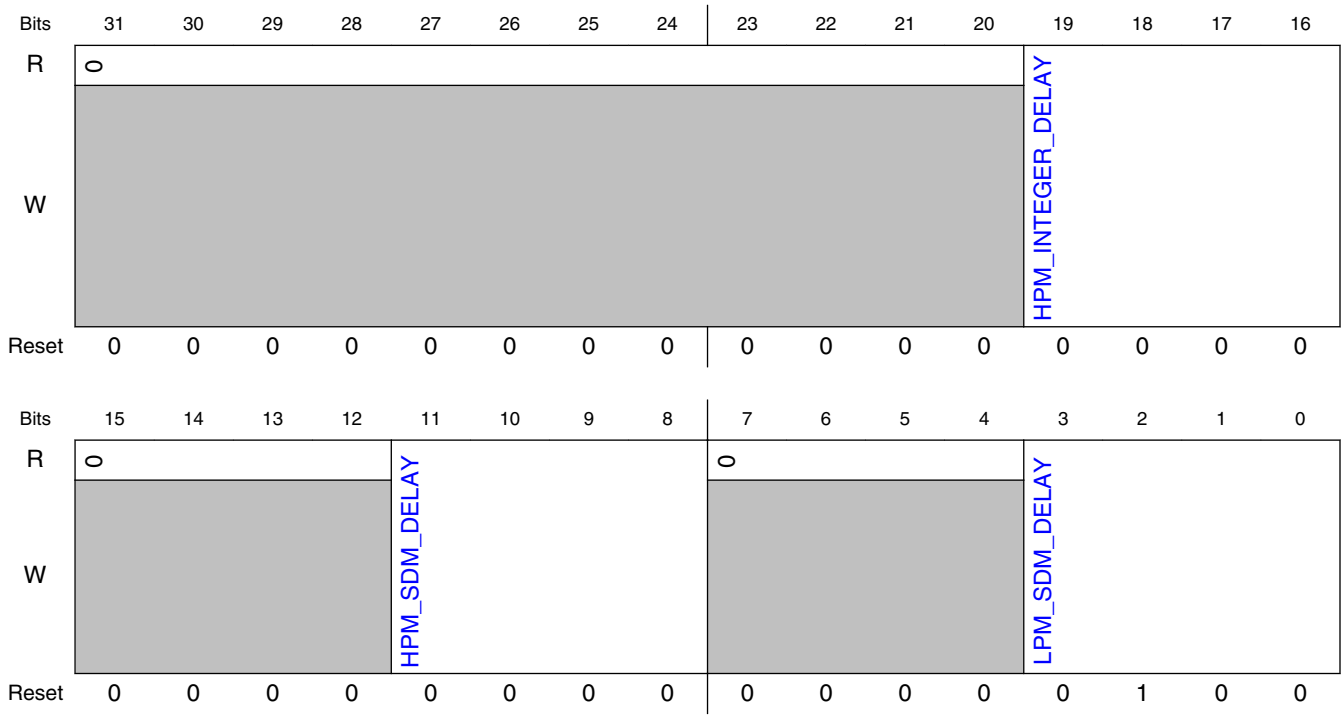
Field	Function
31-28 —	Reserved
27-0 LPM_DENOM_SELECTED	Low Port Modulation Denominator Selected This is the value that is currently being applied to the Low Port Sigma Delta Modulator, for valid Sigma Delta operation the resulting fraction NUM/DENOM must be in the range -0.6 to +0.6

45.3.2.7.1.14 PLL Delay Matching (DELAY_MATCH)

45.3.2.7.1.14.1 Offset

Register	Offset
DELAY_MATCH	3Ch

45.3.2.7.1.14.2 Diagram



45.3.2.7.1.14.3 Fields

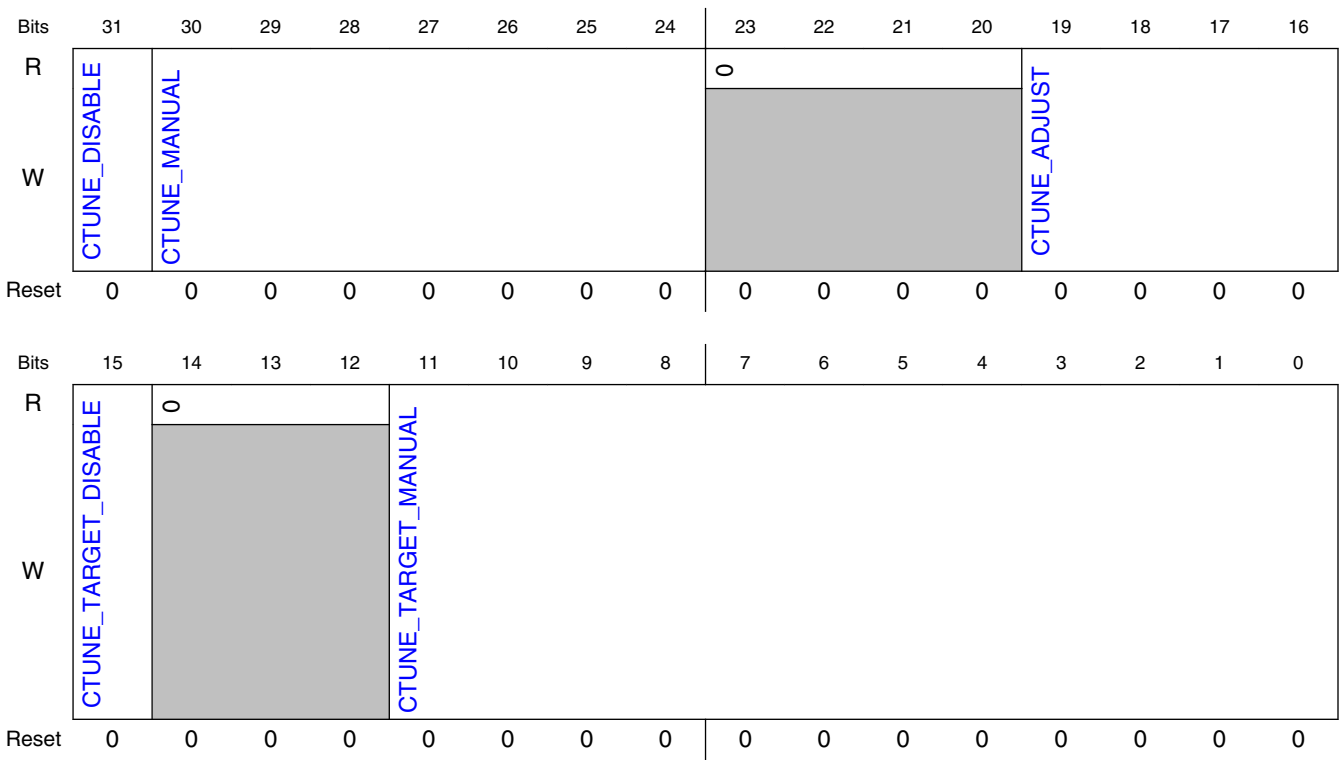
Field	Function
31-20 —	Reserved
19-16 HPM_INTEGER_DELAY	High Port Integer Delay Matching This register selects the number of clock cycles of the PLL Sigma Delta Clock divided by 2 to delay the High Port Integer modulation of the VCO High Port DAC Array.
15-12 —	Reserved
11-8 HPM_SDM_DELAY	High Port SDM Delay Matching 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 Fraction. 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
3-0 LPM_SDM_DELAY	Low Port SDM Delay Matching 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.

45.3.2.7.1.15 PLL Coarse Tune Control (CTUNE_CTRL)

45.3.2.7.1.15.1 Offset

Register	Offset
CTUNE_CTRL	40h

45.3.2.7.1.15.2 Diagram



45.3.2.7.1.15.3 Fields

Field	Function
31 CTUNE_DISABLE	Coarse Tune Disable If this bit is set, the Coarse Tune Setting applied to the VCO comes from the CTUNE_MANUAL register.
30-24 CTUNE_MANUAL	Manual Coarse Tune Setting If CTUNE_DISABLE is set, this register is the value that is applied to the VCO as the Coarse Tune Setting.
23-20 —	Reserved

Table continues on the next page...

Field	Function
19-16 CTUNE_ADJUST	Coarse Tune Count Adjustment This register is a signed three bit value that adjusts the PLL Frequency Meter count used in the Coarse Tune Calibration. The range of adjustment is -8 to +7, and the adjustment is only made to the PLL Frequency count used by the Coarse Tune Calibration sequence.
15 CTUNE_TARGET_DISABLE	Disable Coarse Tune Target If this bit is set, the Frequency Target presented to the Coarse Tune Calibrator comes from the CTUNE_TARGET_MANUAL register.
14-12 —	Reserved
11-0 CTUNE_TARGET_MANUAL	Manual Coarse Tune Target 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.

45.3.2.7.1.16 PLL Coarse Tune Results (CTUNE_RES)

45.3.2.7.1.16.1 Offset

Register	Offset
CTUNE_RES	54h

45.3.2.7.1.16.2 Diagram

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

Bits	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

45.3.2.7.1.16.3 Fields

Field	Function
31-28	Reserved

Table continues on the next page...

Field	Function
—	
27-16 CTUNE_FREQ_SELECTED	Coarse Tune Frequency Selected 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
6-0 CTUNE_SELECTED	Coarse Tune Setting to VCO This is the current VCO Coarse Tune setting, it is the result of the Coarse Tune Calibration, unless overridden using CTUNE_DISABLE.

45.3.3 Receiver Digital Module

45.3.3.1 Introduction

The RX DIG module for Radio 2.1 implements the receive digital signal processing functions common to BTLE, and Generic Link Layer modes. Generally this includes conditioning the input samples from the ADC and produces a data output for the PHY and an RSSI/LQI for the Link Layers.

45.3.3.1.1 Features

The rx_dig module includes the following features:

- Decimation Filter
- I/Q Mismatch Correction
- DC Offset Calibration, Estimation and Correction
- Channel Filter
- Sample Rate Converter
- DC Residual Correction
- AGC & RSSI
- Normalizer
- RC Calibration
- AuxPLL Frequency Calibration

45.3.3.1.2 Modes and operations

The rx_dig module supports BTLE, 802.15.4, and Generic Link Layer modes of operation through use of programmable values such as the oversampling rate, channel filter coefficients, etc. Another configuration is possible to simply pass the values received from the ADC.

The table below shows the data rates supported by RX DIG. Note that the ADC is clocked at twice the reference clock frequency, so ADC clock of 64Mhz corresponds to a 32MHz reference clock, and an ADC clock of 52MHz corresponds to a 26MHz reference clock.

Table 45-18. RX Digital Modes and Rates

ADC clock (MHz)	Decimator OSR	Decimation Filter Output Rate(MHz)	SRC conversion factor	SRC Output Rate (MHz)	Use-Case
64	8	8	bypass	8	BTLE, Generic FSK at 1Mbps
	16	4	bypass	4	IEEE 802.15.4, Generic FSK at 500 kbps
	32	2	bypass	2	Generic FSK at 250 kbps
52	4	13	FOH; 8/13	8	BTLE, Generic FSK at 1Mbps (default)
	6	8.67	ZOH; 12/13	8	BTLE, Generic FSK at 1Mbps (improved blocker rejection but reduced sensitivity)
	8	6.5	FOH; 8/13	4	Generic FSK at 500 kbps(default)
	12	4.33	ZOH; 12/13	4	Generic FSK at 500 kbps (with improved blocker rejection at 3MHz)
	16	3.25	FOH 8/13	2	Generic FSK at 250 kbps
	24	2.167	ZOH; 12/13	2	Generic FSK at 250 kbps

45.3.3.1.3 Block diagram

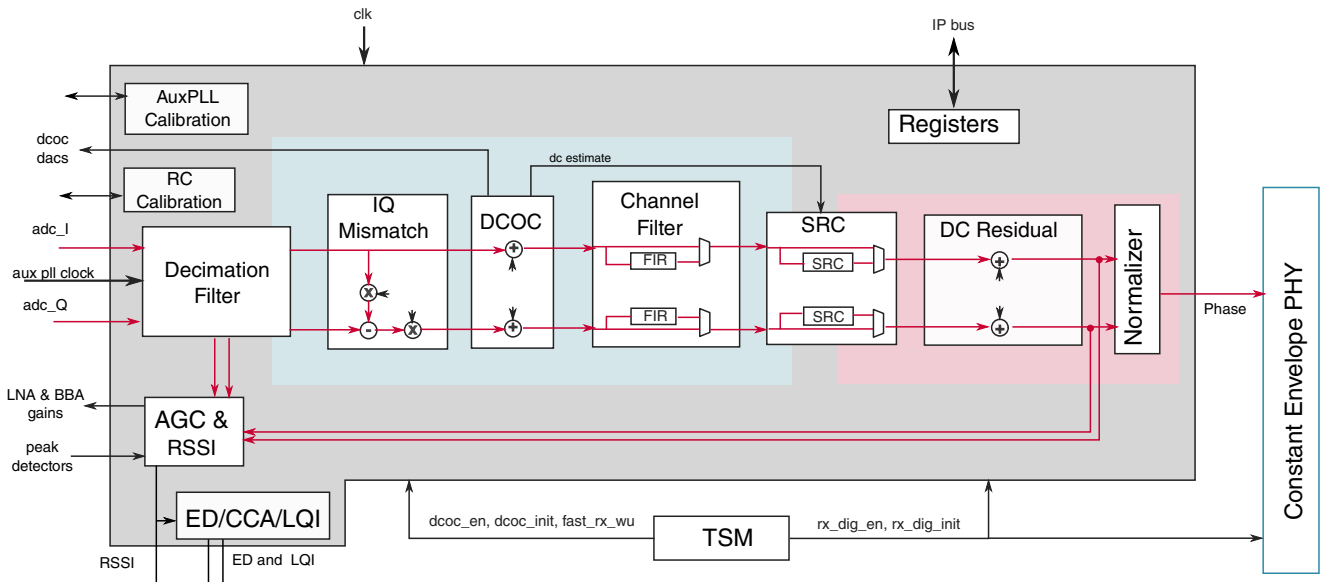


Figure 45-27. RX DIG Block diagram

45.3.3.2 Functional description

The following sections describe functional details of the RX_DIG module.

The RX_DIG module has two overall funtions. Samples from the ADC are computed to usable formats for the demodulator. Additionally, the module conditions the received sample in various ways, including sending feedback to the analog circuitry.

45.3.3.2.1 Decimation Filter

The decimation filter is a 4th order Cascaded Integrator Comb (CIC) filter. Its frequency response comprises a 4th order sinc filter that supports oversampling ratios (OSR) of 4, 6, 8, 12, 16, 24 and 32 . There are two decimation filters: one for the I channel and one for the Q channel. The decimators consume the sigma-delta modulated output from the ADCs at the Aux PLL frequency clock rate (2x reference clock rate) and output samples at the decimation rate in the reference clock domain which used by the rest of the logic in RX DIG. Refer to Table 45-18 for information on the supported decimation rates.

A block diagram of the decimation filter is shown below.

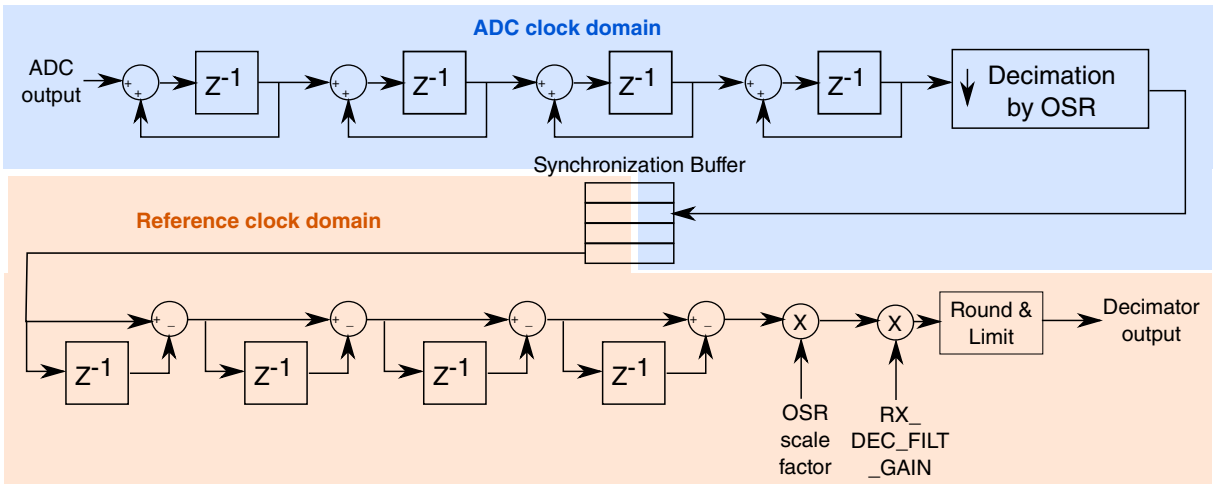


Figure 45-28. Decimation Filter Block Diagram

The decimator output scaling is dependant on the OSR scale factor (fixed in hardware, depends on the programmed OSR) and the programmable RX_DEC_FILT_GAIN. The RX_DEC_FILT_GAIN is intended to be programmed based on the maximum signal level at the ADC input, the ADC gain (1code/1.7V), and the OSR scale factor so that the maximum signal range (+2047/-2048) can be realized in the decimator output. An example of RX_DEC_FILT_GAIN programming is shown in the table below for the case where the maximum input signal level to the ADC is 1.0V. The "Maximum Decimator Output" column is computed as

```
Maximum Decimator Ouput = 1.0V * (1code/1.7V) * OSR^4  * OSR_scale_factor *
RX_DEC_FILT_GAIN / 2
```

Table 45-19. Decimation Filter Gain Table Example, ADC input 1.0V

OSR	Gain at output of 4th order CIC (OSR^4)	OSR scale factor	RX_DEC_FILT_GAIN	Maximum Decimator Output
4	256	2^4	1+1/2+1/8+1/16=1.6875	2033
6	1296	2^2	1+1/4+1/16+1/32=1.3438	2049 (saturated to +2047/-2048)
8	4096	2^0	1+1/2+1/8+1/16=1.6875	2033
12	20736	2^{-2}	1+1/4+1/16+1/32=1.3438	2049 (saturated to +2047/-2048)
16	65536	2^{-4}	1+1/2+1/8+1/16=1.6875	2033
24	331776	2^{-6}	1+1/4+1/16+1/32=1.3438	2049 (saturated to +2047/-2048)
32	1048576	2^{-8}	1+1/2+1/8+1/16=1.6875	2033

The decimator is controlled via several programmable bitfields, and there are also a few status bits , as noted below:

- The OSR is programmed using RX_DEC_FILT_OSR

- The decimation output is scaled based on the OSR and a programmable fractional gain (RX_DEC_FILT_GAIN), as described above. After decimation, the samples are output at the appropriate lower rate in 12-bit signed format.
- The decimator outputs RX_DEC_FILT_SAT_I and RX_DEC_FILT_SAT_Q flags to indicate if the decimator output has saturated during an RX burst
- The single-bit ADC samples can be sampled on either the posedge (default, RX_ADC_NEGEDGE=0) or negedge (RX_ADC_NEGEDGE=1) of the adc_clk.
- The single-bit ADC samples can be mapped as 1'b1->+1, 1'b0->-1 (default, RX_ADC_POL=0) or as 1'b0->+1, 1'b1->-1 (RX_ADC_POL=1).
- The decimator includes a sample buffer to handle the synchronization between the ADC output (adc_clk) clock domain and decimator output (clk) clock domains.
  - The decimator outputs a flag (RX_DEC_FILT_HAZARD) to indicate if a hazard condition has been detected (write and read pointers match) .

For an ADC clocked at a 64 MHz reference clock, an OSR of 8 is used for BTLE and an OSR of 16 is used for IEEE 802.15.4 mode. The frequency response of the decimation filter for these two cases is shown in the figures below.

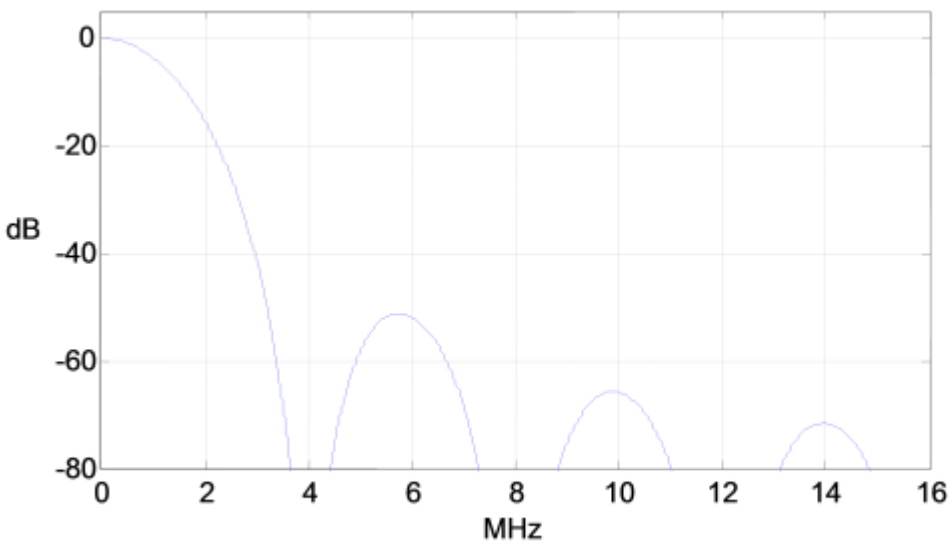


Figure 45-29. 4th order CIC (OSR=8) Decimation Filter Frequency Response

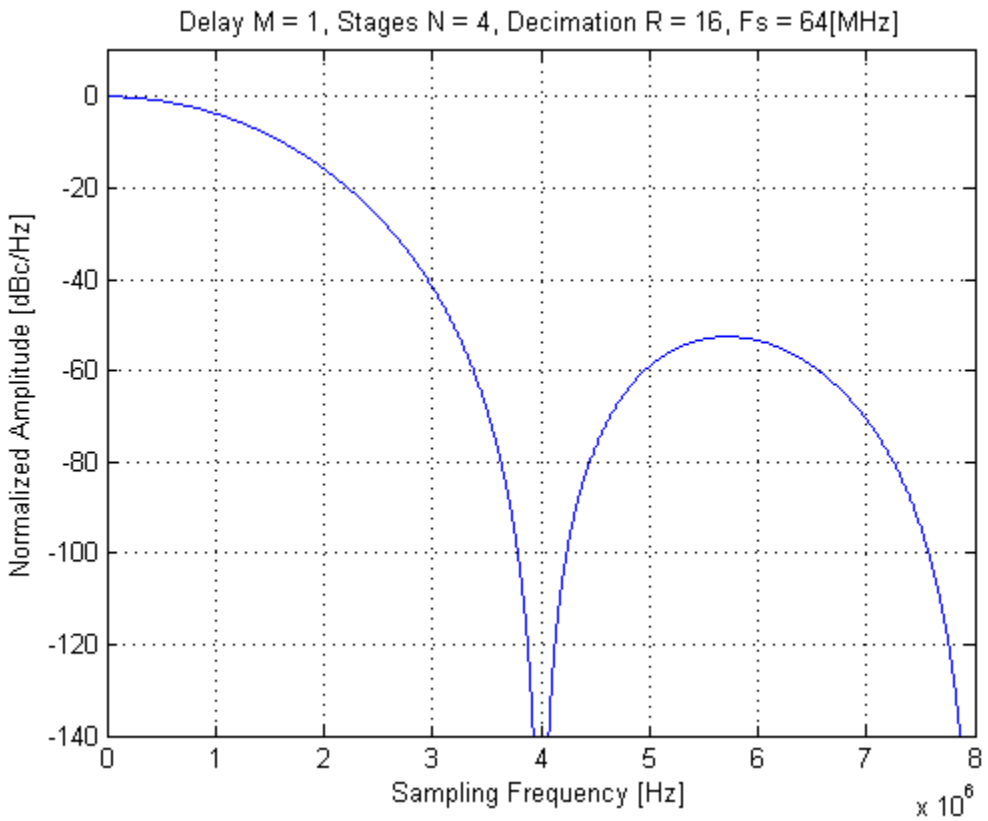


Figure 45-30. 4th order CIC (OSR=16) Decimation Filter Frequency Response

The decimator block provides a separate output for wide-band RSSI measurement that is used by the AGC algorithm and CCA1/CCA3/ED/LQI. This output is at 8MHz (32MHz reference clock) or 8.667MHz (26MHz reference clock). As the objective of the wide-band RSSI measurement is to compute a dBVrms value of the ADC output stream with minimal delay, a 2nd order CIC configuration is used. This 2nd order CIC shares the first two CIC integrators with the main datapath, but uses 2 dedicated difference stages.

The decimator for wide-band RSSI measurements is controlled via several programmable bitfields, and there are also a few status bits , as noted below:

- The decimation ratio is determined by XCVR_CTRL[REF_CLK_FREQ]. For a 32MHz reference clock, the decimation ratio is 8 so the decimator output is 8MHz; for a 26MHz reference clock, the decimation ratio is 6 so the decimation output rate is 8.667MHz
- After decimation, the samples are output at the appropriate lower rate in 10-bit signed format. The decimation output is scaled based on the OSR so that the maximum signal level is +/-576 codes.
- The decimator includes a buffer to handle the synchronization between the ADC output (adc_clk) clock domain and decimator output (clk) clock domains.

- The wideband RSSI decimator outputs a flag (RX_RSSI_FILT_HAZARD) to indicate if a hazard condition has been been detected (write and read pointers match)

45.3.3.2.2 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.

45.3.3.2.2.1 RSSI estimator

The role of the RSSI (Received Signal Strength Indication) estimator is to measure the energy of an incoming signal. The estimator:

- provides a wideband ADC signal level measurement for AGC operation using a decimated version of the ADC output (*RSSI_adc*);
- provides an inband received energy estimate using the "Gain adjustment" block in Figure 45-34;
- targets RSSI accuracy with tunable gains and coefficients.

RSSI accuracy is better than +/- 3dB for input signals in the range of -50dBm to 0dBm. This is the intended operating range for the AGC system. RSSI accuracy is better than +/- 6dB for all input signals below -50dBm.

RSSI estimation is performed using the binary samples output by the ADC. After this operation the signal is filtered using a second order CIC and rate reduction (to 8 MHz if 32 MHz crystal is used or to 8.66MHz for a crystal frequency of 26 MHz). The signal magnitude is then estimated with an L1-norm approach as follows:

$$\textbf{\textit{Magnitude}}(i) = max(|I_i|,|Q_i|) + {}^{3}/_{8} * min(|I_i|,|Q_i|)$$

The estimated magnitude is then passed through a smoothing, low-pass single-tap IIR filter with the transfer function (update factor is α in Figure 45-34):

$$y[n] = (1-\alpha)*y[n-1] + \alpha*x[n]$$

In order to reduce the filter settling duration, when *cca1_ed_trig* is asserted, the α IIR output is seeded with the first value of the estimated magnitude. Configuration of this filter is described in Table 45-22, parameter *rssi_iir_weight*. Then a down-sampler is used to reduce the sampling rate (8 MHz or 8.66 MHz) to a lower rate. A down-sampling

cycle is as follows: *N* samples are accumulated, then the results is divided by *N*, division result is output (at a rate *N* times smaller than the input signal) and the internal accumulator is reset to 0.

Following the smoothing filter and down-sampler, the RSSI estimator converts the approximated magnitude to dB. This conversion is achieved using a look-up table ("lin2dB" block) that approximates:

$$RSSI\_adc = 20*log_{10}(^{approx.\ magnitude}/_{ADC\ full\ scale})$$

The computed value *RSSI_adc* is intended to be used by the AGC mechanism.

RSSI_adc estimate is converted to an estimate for the signal strength at the antenna, in the "Gain Adjustment" block, using parameters as follows:

$$RSSI = RSSI\_adc - lnm\_gain - bbf\_gain + rssi\_adj+13$$

where *rssi_adj* is described in Table 45-22 and represents a constant value which allows taking into account different adjustments needed. The adjustment which accounts for scaled dBVrms to dBm conversion is considered using the constant 13. *lnm_gain* and *bbf_gain* values have resolution of a quarter dB.

The rest of the blocks in Figure 45-34 are used for CCA1, CCA3 and ED. In order to obtain these measurements an averager is used ("RSSI Averager") which essentially is another down-sampler with a factor *M*. This block functions in the same manner as the *N* order down-sampler. In the wideband mode context, the block "RSSI Averager" and the sub-sequent blocks are enabled only if a CCA/ED measurement is in progress or if the test mode is enabled.

45.3.3.2.2.2 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 BBA, and analog LNA. 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_BBA_GAIN_EN and USER_LNA_GAIN_EN bits. If these bits are set, the gains selected in BBA_USER_GAIN and LNA_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 45-31 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 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_LNA_STEP_SZ, AGC_DOWN_BBA_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 LNA_GAIN_SETTLE_TIME and BBA_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 RSSI_adc is quickly checked. If there is not enough ADC headroom (as determined by PRESLOW_DOWN_THRESH), a small downward gain step is taken before proceding to SLOW AGC. If there is too much ADC headroom (as determined by PRESLOW_UP_THRESH), a small upward gain step is taken before proceeding 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 (ipr_agc_fast_expire when in slow mode) 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.

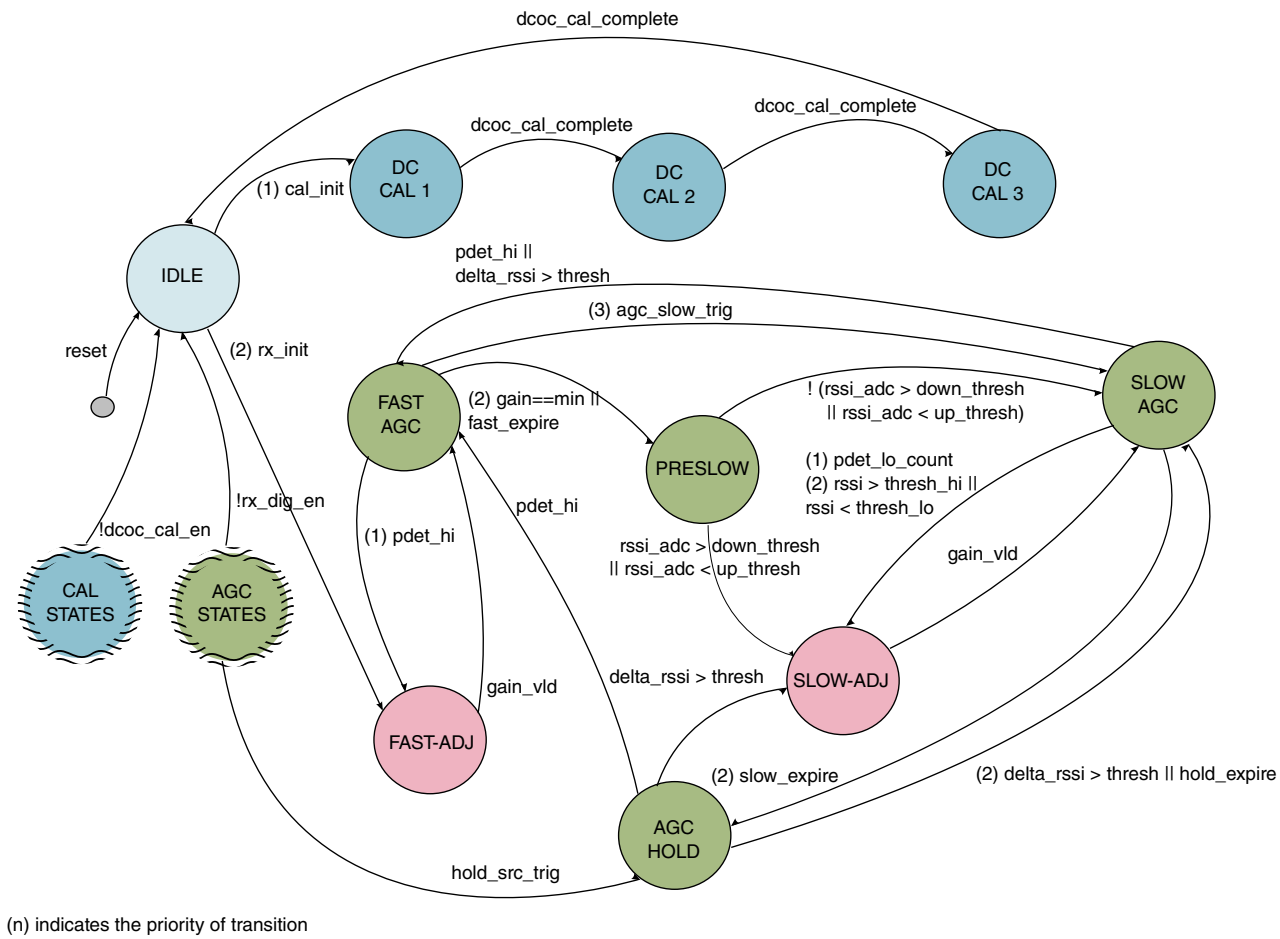


Figure 45-31. AGC State Machine

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 LNA gain and an index for BBA gain. The mapping of these indicies to codes and nominal gains is described in Table 45-20 and Table 45-21. Entry 0 is interpreted as the lowest possible gain setting, and entry 26 the highest.

Table 45-20. LNA Gains

lna_gain (AGC_GAIN_TBL)	LNA Gain (dB)	lna_gain code
0	-8.6	01
1	-3.5	02
2	2.2	03
3	13.9	04
4	19.8	08
5	22.7	0C
6	28.6	18
7	34.4	30
8	39.9	9C
9	45.4	FC

Table 45-21. BBA Gains

bba_gain (AGC_GAIN_TBL)	BBA Gain (dB)	bba_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

45.3.3.2.3 RSSI, Energy Detection, CCA1 and LQI module

There are two modules that achieve RSSI, Energy Detection, CCA, and LQI: rx_agc_rssi_est and rx_cca_ed_lqi.

See the AGC section for additional RSSI details.

The RSSI module is responsible with the generation of estimates used for AGC and RSSI or required by the upper layers of the protocols stack. Two modes are defined:

- **Wideband mode**, which uses as input the samples from the output of the ADC decimator (signals *dec_8m_out_i[10:0]* and *dec_8m_out_q[10:0]*). In this mode the incoming samples have sampling frequency of either 8 MHz (if 32 MHz crystal is used) or 8.66 MHz(if 26 MHz crystal is used).
- **Narrowband mode**, which uses as input the samples from the output of the min-max residual DC offset compensator (signals *narrow_band_out_i[11:0* and *narrow_band_out_q[11:0]*). In this mode, the incoming samples have sampling frequency of either 2, 4 or 8 MHz depending on PHY configuration.

The cca_ed_lqi module uses rssi outputs from the rssi_est module.

Table 45-22. Parameters

<table>
<tr><th>Parameter</th><th>Notes</th></tr>
<tr><td>meas_trans_to_idle</td><td>Flag establishing the state machine (see Figure 45-33) transion following an LQI or CCA1/ED measurement.

<table><tr><th>Value</th><th>Behaviour</th></tr><tr><td>0</td><td>Module transitions to RSSI state</td></tr><tr><td>1</td><td>Module transitions to IDLE state</td></tr></table></td></tr>
<tr><td>cca1_thresh[7:0]</td><td>Threshold used for CCA1 channel state determination expressed in dBm. cca1_thresh is not programmed in RX_DIG, it is programmed in the 802.15.4 link layer.</td></tr>
<tr><td>rssi_iir_weight[3:0]</td><td>Update factor used for IIR filtering of L1 magnitude and noise (α in Figure 45-34 and Figure 45-35).

<table><tr><th>Bit-field value</th><th>α</th></tr><tr><td>0 (bypass)</td><td>1</td></tr><tr><td>1</td><td>1/2</td></tr><tr><td>2</td><td>1/4</td></tr><tr><td>3</td><td>1/8</td></tr><tr><td>4</td><td>1/16</td></tr><tr><td>5</td><td>1/32</td></tr><tr><td>6-15</td><td>Reserved</td></tr></table></td></tr>
<tr><td>rssi_n_window_nb[2:0]</td><td>Downsampling factor for filtered magnitude (N in Figure 45-34 and Figure 45-35).

<table><tr><th>Bit-field value</th><th>N</th></tr><tr><td>0 (bypass)</td><td>1</td></tr><tr><td>1</td><td>2</td></tr><tr><td>2</td><td>4</td></tr></table></td></tr>
</table>

Table continues on the next page...

Table 45-22. Parameters (continued)

<table>
<tr><th>Parameter</th><th>Notes</th></tr>
<tr><td></td><td>
<table>
<tr><th>Bit-field value</th><th>N</th></tr>
<tr><td>3</td><td>8</td></tr>
<tr><td>4</td><td>16</td></tr>
<tr><td>5</td><td>32</td></tr>
</table>
</td></tr>
<tr><td>rssi_noise_avg_factor[2:0]</td><td>Factor used for RSSI and SNR averaging (M in Figure 45-34 and Figure 45-35).
<table>
<tr><th>Bit-field value</th><th>M</th></tr>
<tr><td>0 (bypass)</td><td>1</td></tr>
<tr><td>1</td><td>64</td></tr>
<tr><td>2</td><td>70</td></tr>
<tr><td>3</td><td>128</td></tr>
<tr><td>4</td><td>139</td></tr>
<tr><td>5</td><td>256</td></tr>
<tr><td>6</td><td>277</td></tr>
<tr><td>7</td><td>512</td></tr>
</table>
</td></tr>
<tr><td>rssi_noise_avg_delay[5:0]</td><td>Programmable delay used to delay the enable of averagers "RSSI Averager" and "Noise Averager" after a cca1_ed_trig assertion (in wideband mode) or after a aa_sfd_matched assertion (in narrowband mode). The delay is expressed in ticks of frequency Fs/N, where Fs is the ADC decimator output sampling frequency in wideband mode or the base-band sampling frequency in the narrowband mode.</td></tr>
<tr><td>lqi_corr_thresh[7:0]</td><td>Threshold used to compare corr_mag (see Figure 45-35).</td></tr>
<tr><td>corr_cnt_thresh[7:0]</td><td>Threshold used to compare the counted correlation magnitudes exceeding lqi_corr_thresh (see Figure 45-35).</td></tr>
<tr><td>lqi_rssi_sens[3:0]</td><td>Unsigned integer used for calculation of LQI Sensitivity, through relation:
Sensitivity = -103+lqi_rssi_sens.</td></tr>
<tr><td>lqi_rssi_weight[2:0]</td><td>RSSI weight used for LQI calculation, w_{RSSI}.
<table>
<tr><th>Bit-field value</th><th>w_{RSSI}</th></tr>
<tr><td>0</td><td>2.0</td></tr>
<tr><td>1</td><td>2.125</td></tr>
<tr><td>2</td><td>2.25</td></tr>
<tr><td>3</td><td>2.375</td></tr>
<tr><td>4</td><td>2.5</td></tr>
<tr><td>5</td><td>2.625</td></tr>
<tr><td>6</td><td>2.75</td></tr>
<tr><td>7</td><td>2.875</td></tr>
</table>
</td></tr>
<tr><td>rssi_adj[7:0]</td><td>RSSI calculation adjustment (8-bit signed with quarter dB resolution).</td></tr>
</table>

Table continues on the next page...

Table 45-22. Parameters (continued)

<table>
<tr><th>Parameter</th><th>Notes</th></tr>
<tr><td>snr_lqi_weight[3:0]</td><td>SNR weight used for LQI calculation, w_{SNR}.

<table><tr><th>Bit-field value</th><th>w_{SNR}</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr><tr><td>2</td><td>1.125</td></tr><tr><td>3</td><td>1.250</td></tr><tr><td>4</td><td>1.375</td></tr><tr><td>5</td><td>1.500</td></tr><tr><td>6</td><td>1.625</td></tr><tr><td>7</td><td>1.750</td></tr><tr><td>8</td><td>1.875</td></tr><tr><td>9</td><td>2.000</td></tr><tr><td>10</td><td>2.125</td></tr><tr><td>11</td><td>2.250</td></tr><tr><td>12</td><td>2.375</td></tr><tr><td>13</td><td>2.500</td></tr><tr><td>14</td><td>2.625</td></tr><tr><td>15</td><td>2.750</td></tr></table></td></tr>
<tr><td>snr_adj[5:0]</td><td>SNR calculation adjustment (6-bit signed with half dB resolution).</td></tr>
<tr><td>lqi_bias[3:0]</td><td>Bias used for LQI calculation, LQI_{bias} using formula:
LQI_{bias} = -36 + 3*lqi_bias.</td></tr>
</table>

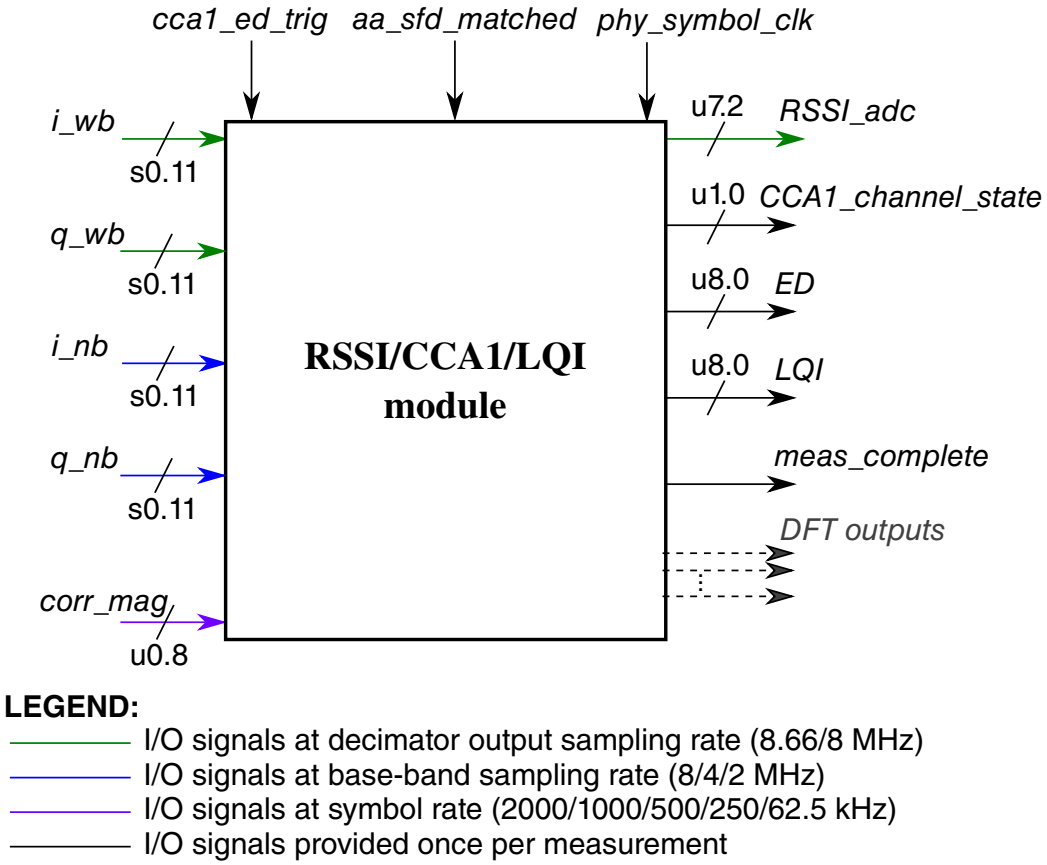


Figure 45-32. Top level view of the block diagram

Following table is showing the valid configurations and the valid outputs thereto.

Table 45-23. Valid configurations and corresponding outputs

cca1_ed_trig	*sfd_aa_match*	State	Valid outputs
0	0	RSSI	*RSSI_adc*
1	0	CCA1/ED measurement	*RSSI_adc* *zb_cca1_channel_state* *zb_cca3_channel_state* *ED*
0	1	LQI measurement	*LQI*
1	1	LQI measurement	*LQI*

Following figure describes the state machine of the module.

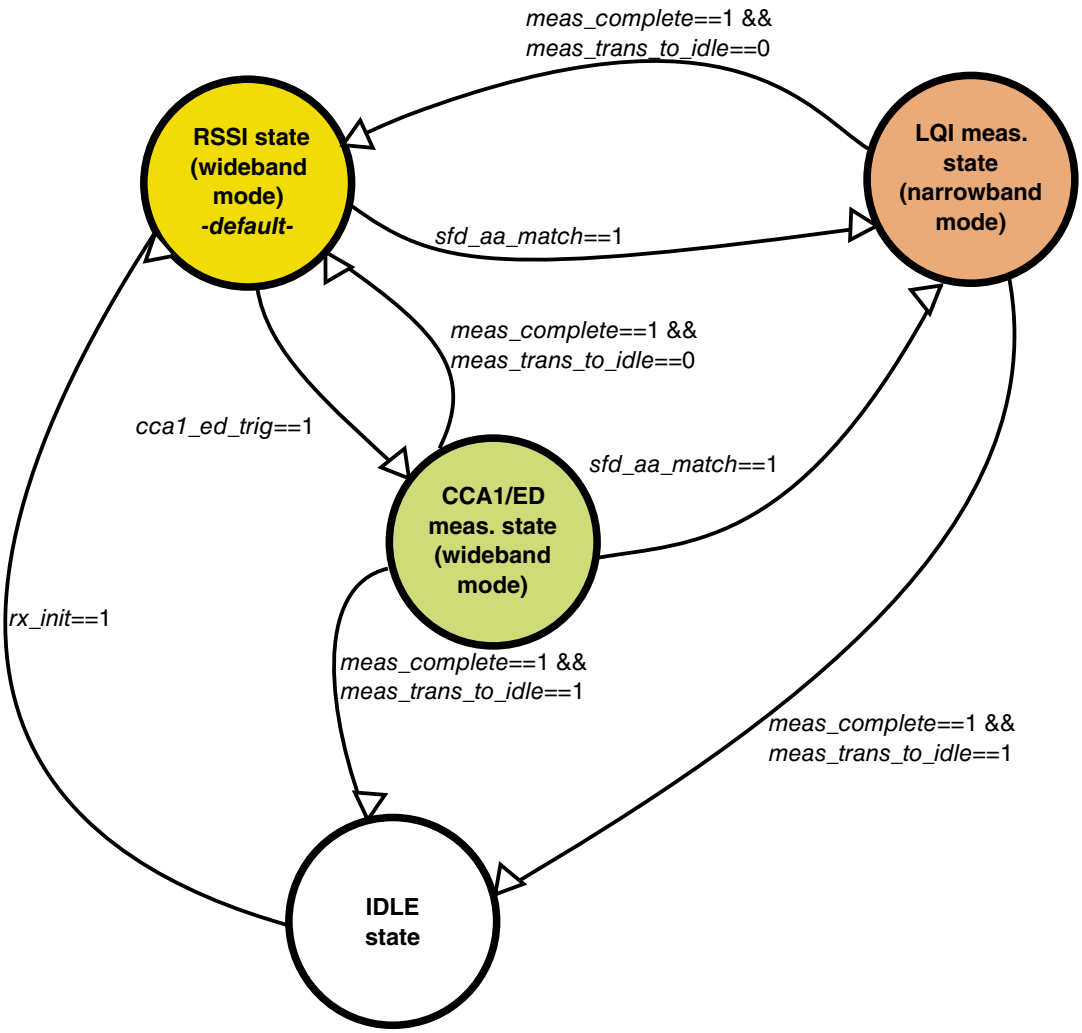


Figure 45-33. Module state machine

45.3.3.2.3.1 Wideband mode

The instantiation of the block diagram corresponding to wideband mode is illustrated in Figure 45-34. Blocks drawn using dashed lines are shared between the wideband and narrowband modes. Following functions are intended in the wideband mode (as described in Table 45-23):

- Estimation of RSSI at the output of the ADC decimator (*RSSI_adc*) used for AGC and of received inband RSSI;
- Energy detection (*ED*), required by the upper layers, which is an 8 bits value achieved through measuring the energy of the channel;
- Clear Channel Assessment Mode 1 (*zb_cca1_channel_state*), required by the upper layers, which is a flag indicating whether the channel is busy or idle.

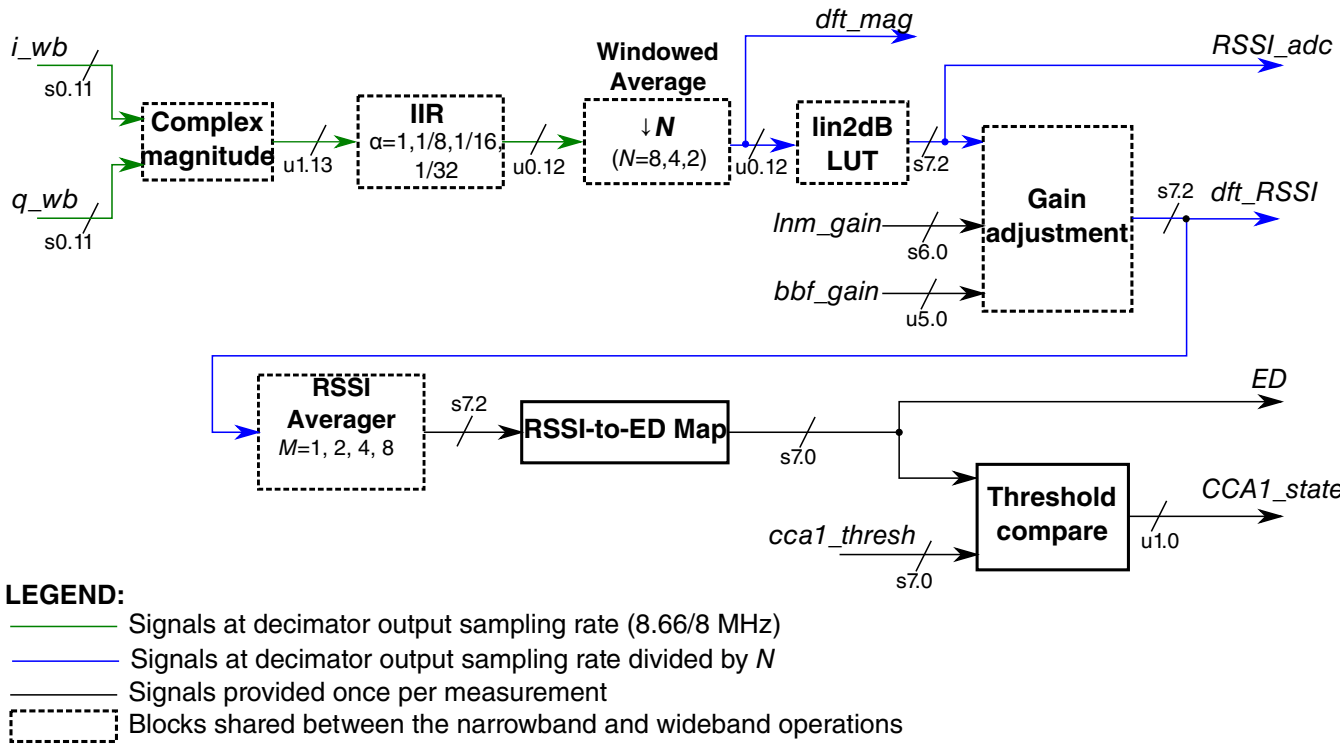


Figure 45-34. Block diagram instantiated for wideband mode

45.3.3.2.3.1.1 ED and CCA1

Module transitions to CCA1/ED measurement state and ED and CCA1 operations start based on a command signal *cca1_ed_trig* (configuration described in Table 45-23). When this signal is asserted, signal *meas_complete* is forced to low state.

When a *cca1_ed_trig* command is asserted, "RSSI averager" is enabled after a programmable delay in order to take into consideration the settling time of the IIRs (programmable delay is configured via parameter *rssi_noise_avg_delay*). After a result is produced by the "RSSI averager" block (i.e. after *M*N* samples have been input to the RSSI/CCA/ED/LQI module since "RSSI averager" has been enabled) in Figure 45-34 signal *meas_complete* is asserted. This signal is intended to let know the external modules consuming the outputs that ED, CCA1 and CCA3 values are ready. In a non-test mode, the module returns to RSSI state after asserting *meas_complete* signal, if parameter *meas_trans_to_idle* is 0 or it transitions to IDLE state if parameter *meas_trans_to_idle* is 1 (see Table 45-22). The measured values of ED, CCA1 and CCA3 are stored until the next deassertion of signal *meas_complete*.

However, the module may be configured to remain in CCA1/ED measurement state for an arbitrary period of time in a test scenario.

Table 45-24. Typical CCA/ED measurement window values and corresponding configuration

Crystal frequency [MHz]	ADC decimator output sampling frequency [MHz]	*N*	*M*	Approximative duration [µs]
32	8	16	64	128
26	8.66	16	70	128
32	8	8	128	128
26	8.66	8	139	128
32	8	8	64	64
26	8.66	8	70	64
32	8	4	64	32
26	8.66	4	70	32

Based on computed value *RSSI* an energy detection value *ED* is computed, expressed in dBm. This value is obtained by rounding the output of the "RSSI Averager" block to 8 signed bits by using block "RSSI-to-ED Map" in Figure 45-34.

Assessment of the channel's state is by comparing value *ED* with threshold *cca1_thresh*. If the threshold is exceeded at the end of the measurement then *zb_cca1_channel_state* is put to '1'. Otherwise it is put to '0'.

The CCA3 estimation *zb_cca3_channel_state* is realized by combining *zb_cca1_channel_state* and *zb_cca2_channel_state* . This is handled outside RX_DIG.

45.3.3.2.3.2 Narrowband mode

The instantiation of the block diagram corresponding to narrowband mode is illustrated in Figure 45-35. Blocks drawn using dashed lines are shared between the wideband and narrowband modes. Following functions are intended for this mode (as described in Table 45-23):

- Link Quality Indication determination (*LQI*), required by the upper layers. LQI is an 8 bits value obtained by combining inband RSSI (corresponding to channel filtered samples), SNR (signal-to-noise ratio) and correlation magnitudes;

LQI is computed in three steps:

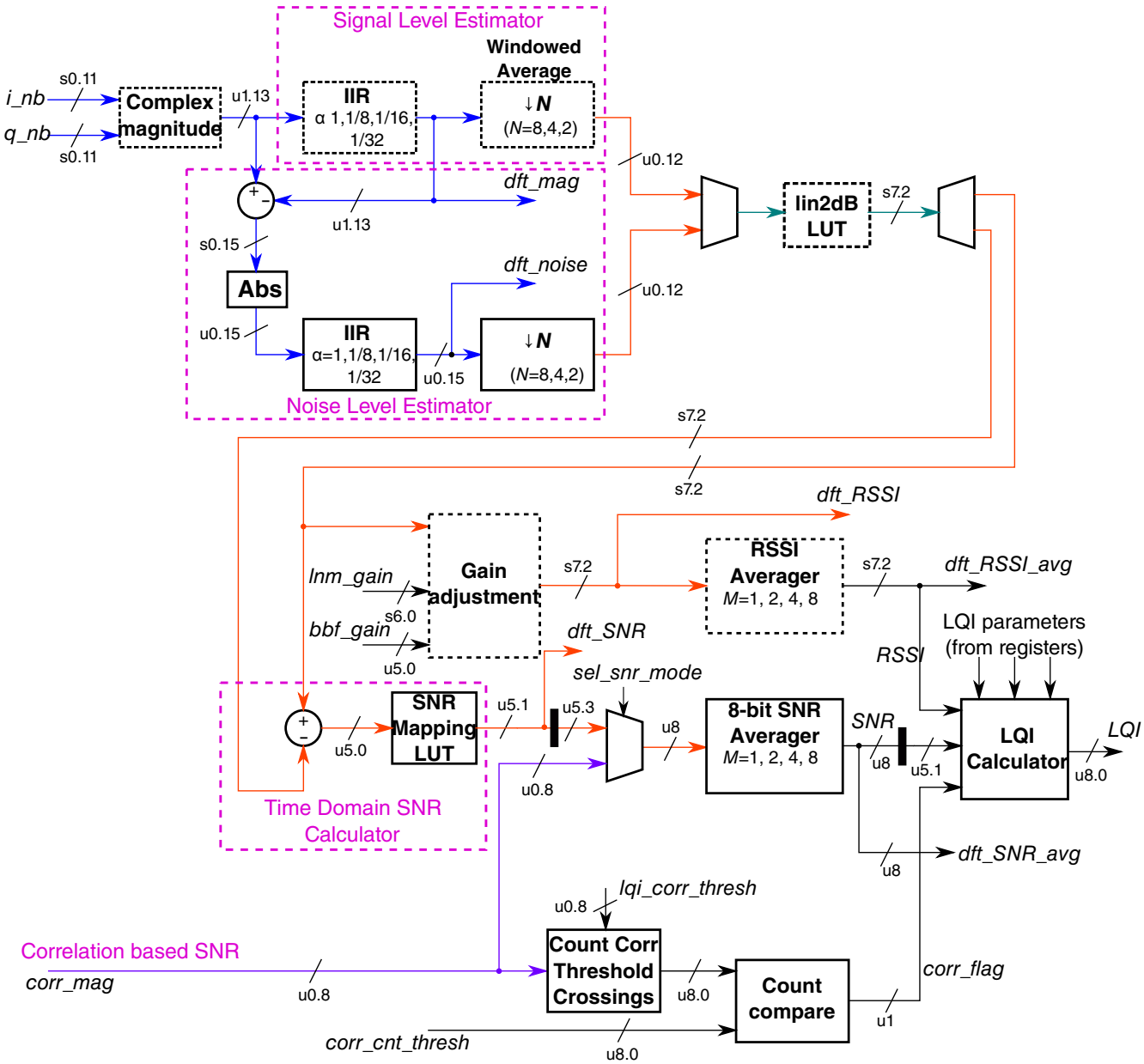
- RSSI (using narrow band I/Q samples) and SNR estimation;
- Correlation flag determination, used to determine whether receiver is at the sensitivity boundary or not;
- LQI calculation which represents a linear combination of RSSI, SNR and correlation flag.

An LQI measurement is triggered by the assertion of *aa_sfd_matched* signal. Following this event, the module transitions to LQI measurement state. When *aa_sfd_matched* is asserted the signal *meas_complete* is deasserted (i.e. forced to low state). The blocks "RSSI averager" and "Noise averager" are enabled after a programmable delay in order to take into consideration the settling time of the IIRs (programmable delay is configured via parameter *rssi_noise_avg_delay*). Values *N* and *M* (configured through parameters *mag_downsamp_factor* and *rssi_noise_avg_factor* respectively), and the base-band sampling frequency *Fs*, directly determine the LQI measurement window duration. At the end of this window *meas_complete* is asserted and measured value *LQI* is provided, after a result is produced by the "RSSI averager" block (i.e. after *M*N* samples have been input to the RSSI/CCA/ED/LQI module since "RSSI averager" has been enabled in Figure 45-34). This signal is intended to let know the external modules consuming the outputs that LQI value is ready. The measured value of LQI is stored until the next deassertion of signal *meas_complete*. In a non-test mode, the module returns to RSSI state after asserting *meas_complete* signal, if parameter *meas_trans_to_idle* is 0 or it transitions to IDLE state if parameter *meas_trans_to_idle* is 1 (see Table 45-22).

However, the module may be configured to remain in LQI measurement state for an arbitrary period of time in a test scenario.

Table 45-25. Examples of configuration of LQI block

Protocol	*N*	*M*	*Fs* [MHz]	Window duration [µs]
IEEE 802.15.4	4	128	4	128
BLE	8	32	8	32
BLE	8	64	8	64
FSK-500kbps	4	64	2	128
FSK- 250kbps	4	64	2	128



LEGEND:

- Signals at base-band sampling rate F_S (8/4/2 MHz)
- Signals at F_S/N
- Signals at $2*(F_S/N)$
- Signals at symbol rate (2000/1000/500/250/62.5 kHz)
- Signals provided once per measurement
- Blocks shared between the narrowband and wideband operations

Figure 45-35. Block diagram instantiated for narrowband mode

45.3.3.2.3.2.1 RSSI and SNR estimation

RSSI is estimated in a similar manner as described in RSSI estimator. The downsampling with *N* block is basically an accumulator. At the end of *N* accumulated values, it divides the result by *N*, outputs the result and resets the internal accumulator to 0. In the same manner function the blocks "RSSI Averager" and "SNR Averager" in Figure 45-35.

In order to estimate the SNR, a noise standard deviation estimation is needed. Noise is estimated after taking the absolute value of the subtraction of smoothed magnitude from the magnitude samples. After IIR and downsampling by *N* of magnitude and noise, conversion from linear domain to dBm is needed. In order to save die-size, "lin2dB" block is shared between the two processing paths by using a synchronous pair MUX/DEMUX. The sampling rate at the input of the "lin2dB" block is doubled due to muxing.

The signal magnitude and noise processing paths are outputting estimates of magnitude and noise respectively, expressed in dBm. The SNR is estimated by subtracting noise from signal. However, an estimation error is present due to the fact that signal and noise are estimated by averaging L1 magnitude samples and not energy values. In order to compensate for this error a non-linear SNR mapping look-up table (block "SNR Mapping LUT") is used in order to map measured SNR to more accurate SNR values. LUT values are specified in Table 45-26. These values have 0.5 dB resolution.

Table 45-26. SNR mapping LUT

Input value (format (u,5,0))	Output value (format (u, 5,1))	Input value (format (u,5,0))	Output value (format (u, 5,1))
{0,1,2,...,8}	0	17	12
9	1	18	13.5
10	3	19	15
11	4.5	20	16
12	6	21	17.5
13	7.5	22	19.5
14	8.5	23	21.5
15	10	24	25
16	11	{25,26,27,..,31}	26.5

The estimated SNR is adjustable by addition to the estimated value of the value of the parameter *snr_adjustment_factor* (signed integer) in Table 45-22.

After estimating RSSI and SNR the two quantities are passed through the "RSSI Averager" and "SNR Averager" blocks respectively.

45.3.3.2.3.2.2 Correlation flag determination

During the LQI measurement window, a counter is used to count how many times the correlation products, provided at symbol rate, exceed the correlation threshold *lqi_corr_thresh*. The count is compared with threshold *corr_cnt_thresh*. *corr_cnt_thresh* takes values between 0 and the maximum number of symbols which can be received within the measurement window.

Table 45-27. Examples of maximum number of symbols in the measurement window

Protocol	Symbol rate [kHz]	Window duration [µs]	Maximum *corr_cnt_thresh*
IEEE 802.15.4	62.5	128	8
BLE	1000	32	32
FSK-500kbps	500	64	32

The output of the comparator, *corr_flag*, will take value 0 if *corr_cnt_thresh* is exceeded and 1 otherwise.

45.3.3.2.3.2.3 LQI calculator

Based on narrow band *RSSI* (expressed in dBm), *SNR* (dB) and *corr_flag*, *LQI_raw* is computed as described in following formula:

$$LQI\_raw = (RSSI\text{-}Sensitivity)*w_{RSSI} + SNR*w_{SNR} - corr\_flag*w_{CorrFlag}+LQI_{bias}$$

Configuration of parameters *Sensitivity*, w_{RSSI}, w_{SNR} and LQI_{bias} is described in Table 45-22. $w_{CorrFlag}$ is a constant and takes value 200.

After computing *LQI_raw*, the obtained value is saturated to interval [0,255] and then quantized using 8 integer bits. Thus *LQI* is determined.

45.3.3.2.4 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.

During factory calibration, the I/Q phase adjustment coefficient (iqmc_phase_adj) and the I/Q gain adjustment coefficient (iqmc_gain_adj) are determined. 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 16). 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. The gain calibration and phase calibration loops are shown in the figures below.

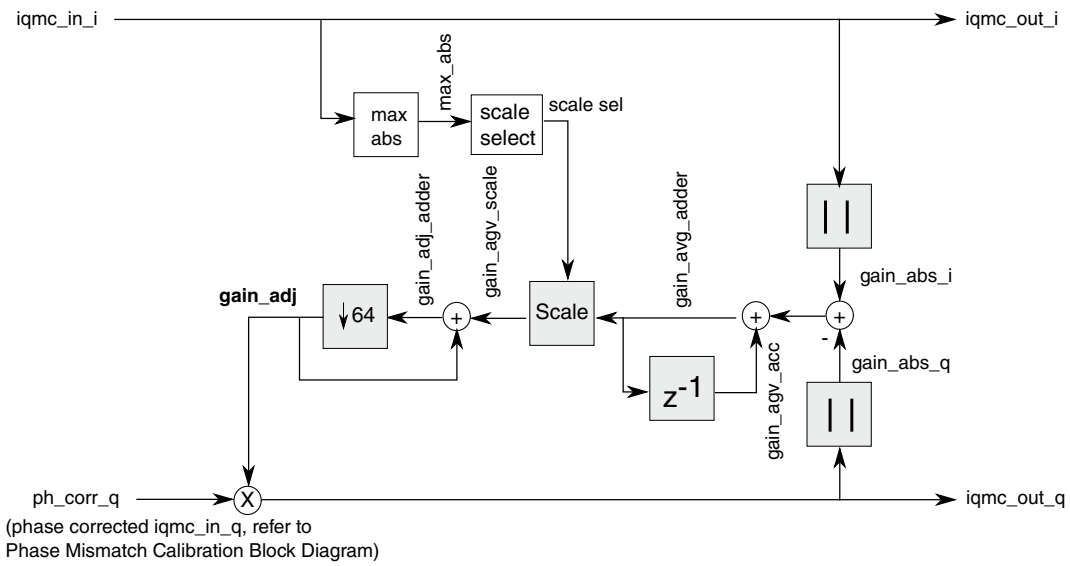


Figure 45-36. I/Q Gain Mismatch Calibration Block Diagram

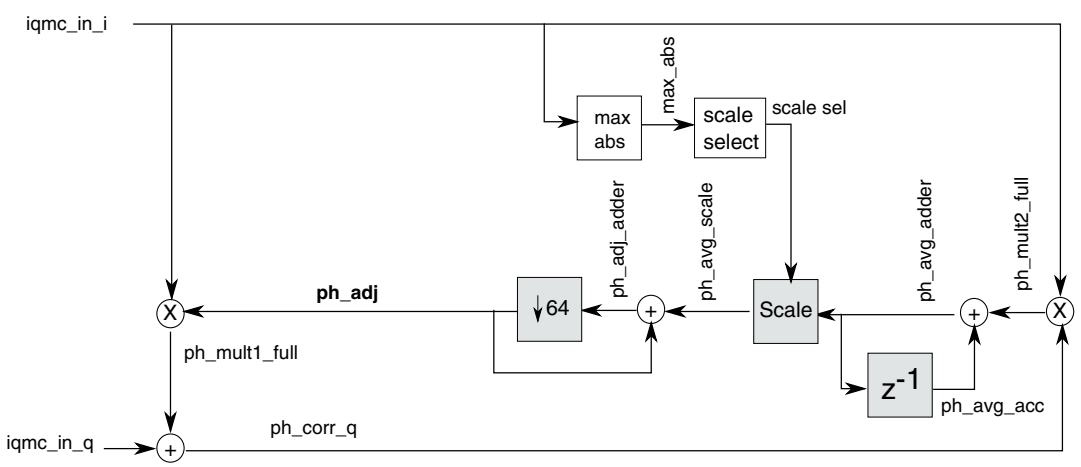


Figure 45-37. I/Q Phase Mismatch Calibration Block Diagram

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 (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. There is no hardware calibration support to determine iqmc_dc_gain_adj.

The gain and phase corrections are illustrated in the figure below.

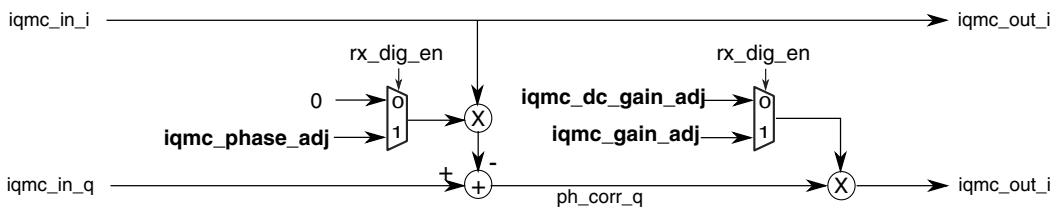


Figure 45-38. I/Q Mismatch Correction Block Diagram

45.3.3.2.5 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 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 receiver, there are three distinct DC offset correction points, which are at the TZA output, BBA output (after first BBA 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 BBA DC corrections are stored in the AGC gain lookup table.
- During AGC, the system applies a DC offset correction corresponding to the {LNA, BBA} 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.

A block diagram of the RX digital DCOC is shown in the figure below.

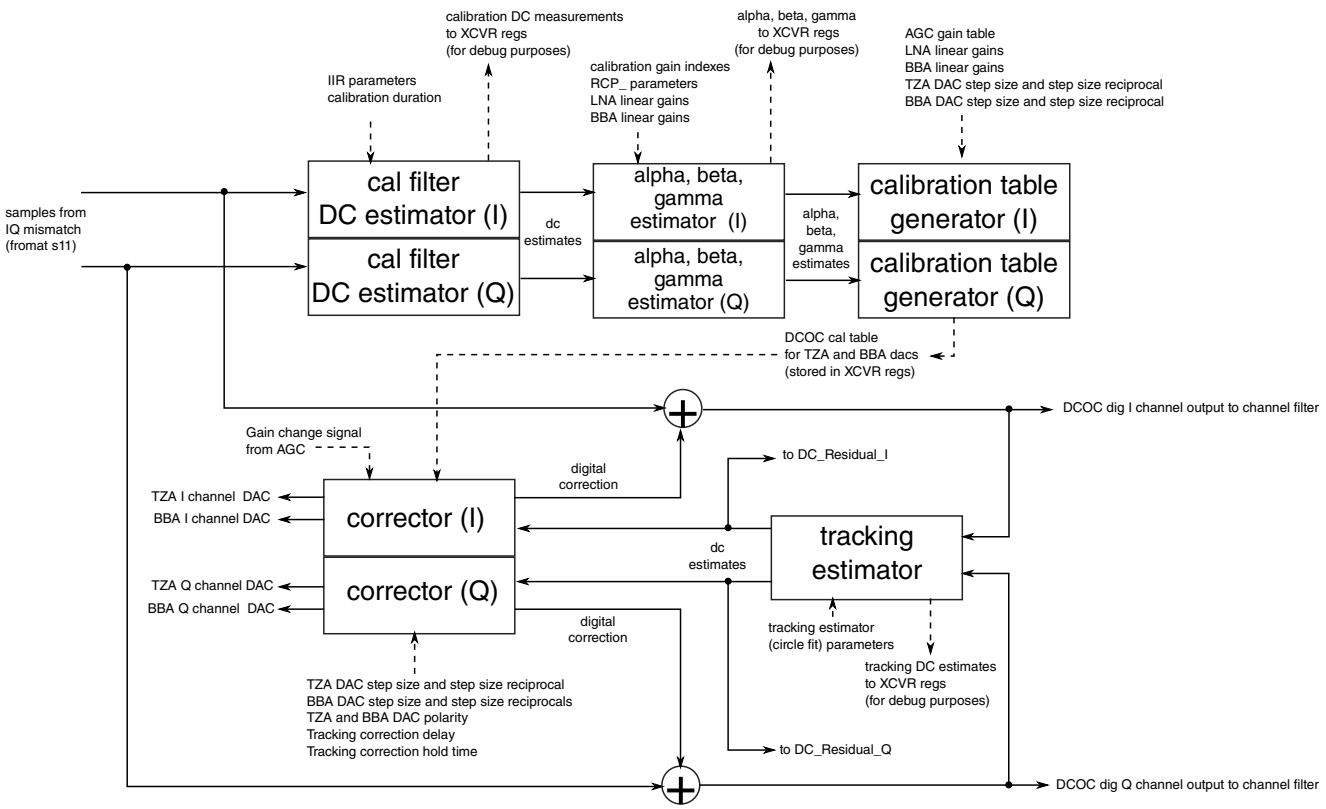


Figure 45-39. RX_DIG DCOC Block Diagram

45.3.3.2.5.1 Operating Modes

The DCOC supports several programmable bits to control the DCOC behavior. The combinations of these control bits which are supported are described in the table below.

Table 45-28. DCOC Operating Modes

Mode	RX_ DCOC_EN	RX_DCOC _CAL_EN	CORRECT _EN	CORRECT _SRC	TRK_ FROM_0	TRK_EST _OVR	MAN
Calibrate and Correct with Tracking	1	1	1	1	0 or 1	0 or 1	0

Table continues on the next page...

Table 45-28. DCOC Operating Modes (continued)

Mode	RX_ DCOC_EN	RX_DCOC _CAL_EN	CORRECT _EN	CORRECT _SRC	TRK_ FROM_0	TRK_EST _OVR	MAN
Calibrate and Correct without Tracking	1	1	1	0	X	0 or 1	0
Correct with Tracking (no calibration)	1	0	1	1	0 or 1	0 or 1	0
Correct without Tracking (no calibration)	1	0	1	0	X	0 or 1	0
Debug: Calibration only (no correction, no tracking).	1	1	0	X	X	0 or 1	0
Debug: Manual mode (manual DAC values and DCOC_DIG_MAN corrections applied). No calibration, no correction	X	0	X	X	X	X	1
Debug: Tracking estimator enabled; manual DAC values applied. No calibration, no correction	1	0	0	X	0 or 1	1	0
DCOC disabled. Manual DAC values applied. No calibration, no correction	0	0	X	X	X	X	0

45.3.3.2.5.2 DCOC State Diagram

The main states of the DCOC are described in the bullet list and figure below.

- WAIT (IDLE)
- CALIBRATION. State where DCOC is performing calibration measurements, estimating alpha/beta/gamma, and starts writing the DCOC OFFSET table
- APPLY_IDX_CAPT. This state is also associated with calibration. DCOC transitions to this state from CALIBRATION when the table index matches the AGC's starting index. The DCOC values matching this index are applied to the DCOC DACs, and DCOC then finishes writing the rest of the DCOC OFFSET table
- NO_CAL_INIT. State where no calibration is performed, but correction is enabled. In this state, the DCOC will output the DCOC DAC values from the DCOC OFFSET table corresponding to the AGC's starting index.
- CORRECT. State where the AGC is enabled and DCOC changes the DCOC DAC values when AGC gain changes by reading from the DCOC OFFSET table.
- CORRECT_AND_TRACK. Similar to CORRECT state, but the DCOC tracking estimator is also activated, to further refine the DCOC DAC values after an AGC gain change, and also to apply a digital correction to the DCOC output.

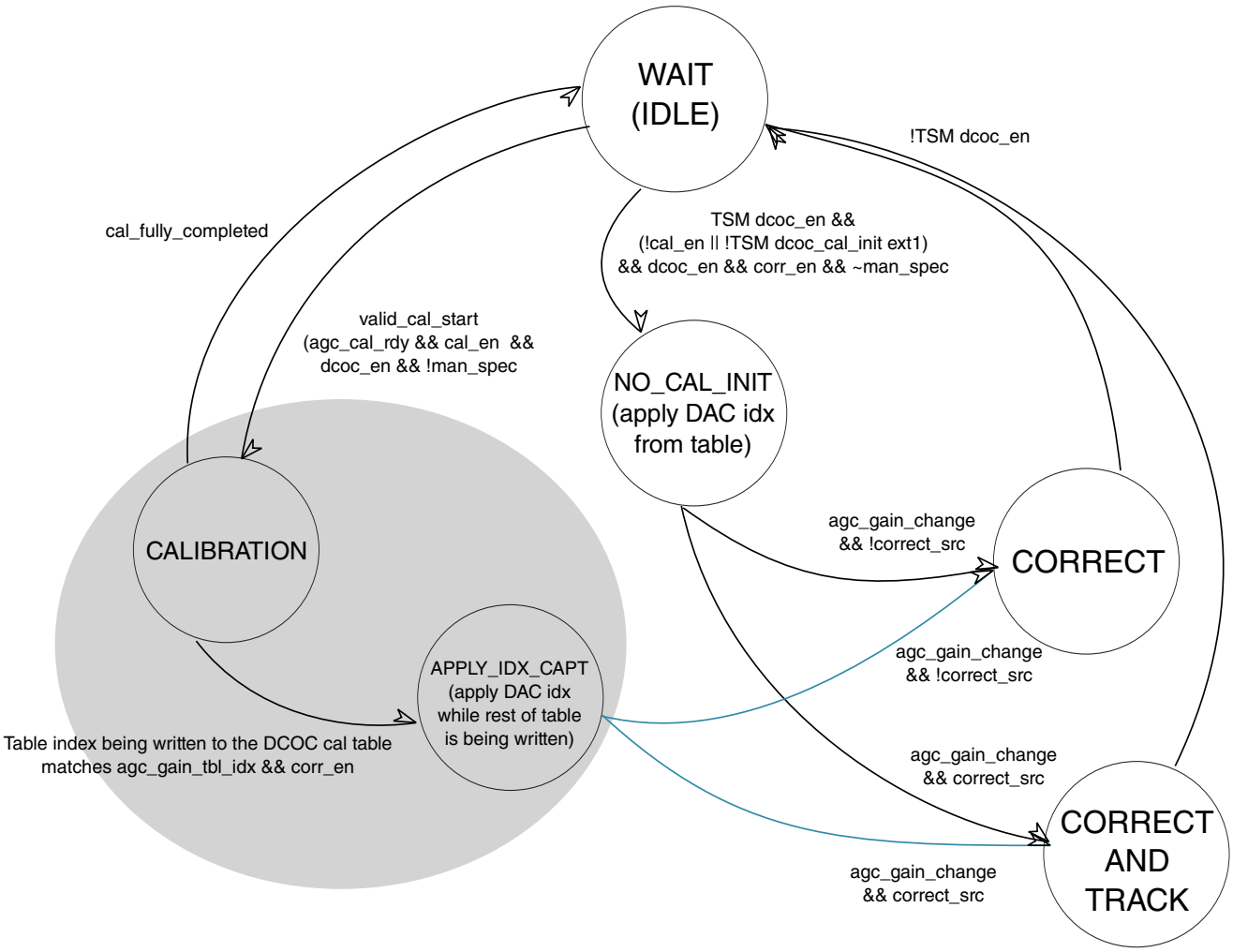


Figure 45-40. DCOC State Diagram

45.3.3.2.5.3 DC Offset Calibration

If the RX_DIG_CTRL[RX_DCOC_EN] and RX_DIG_CTRL[RX_DCOC_CAL_EN] bits are set, the 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 (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*DCOC_CTRL_0[DCOC_CAL_DURATION] + (4*27 + 15)/
reference_clock_frequency
```

A value of DCOC_CAL_DURATION=19 and a reference clock frequency of 32MHz results in a calibration time of 60.8us. The system will operate correctly as long as the calibration completes before the 1us TSM RX_INIT pulse completes.

The figure below illustrates the DC Offset Calibration timing. To allow as much time as possible for the DACs to settle before the AGC is enabled, the DCOC will initialize the DACs as soon as it calculates the DCOC table entry corresponding to the initial AGC table index. Normally, the AGC table index begins at 26 (as shown in the figure below), so the DCOC always calculates and writes the DCOC calibration table from index 26 to index 0.

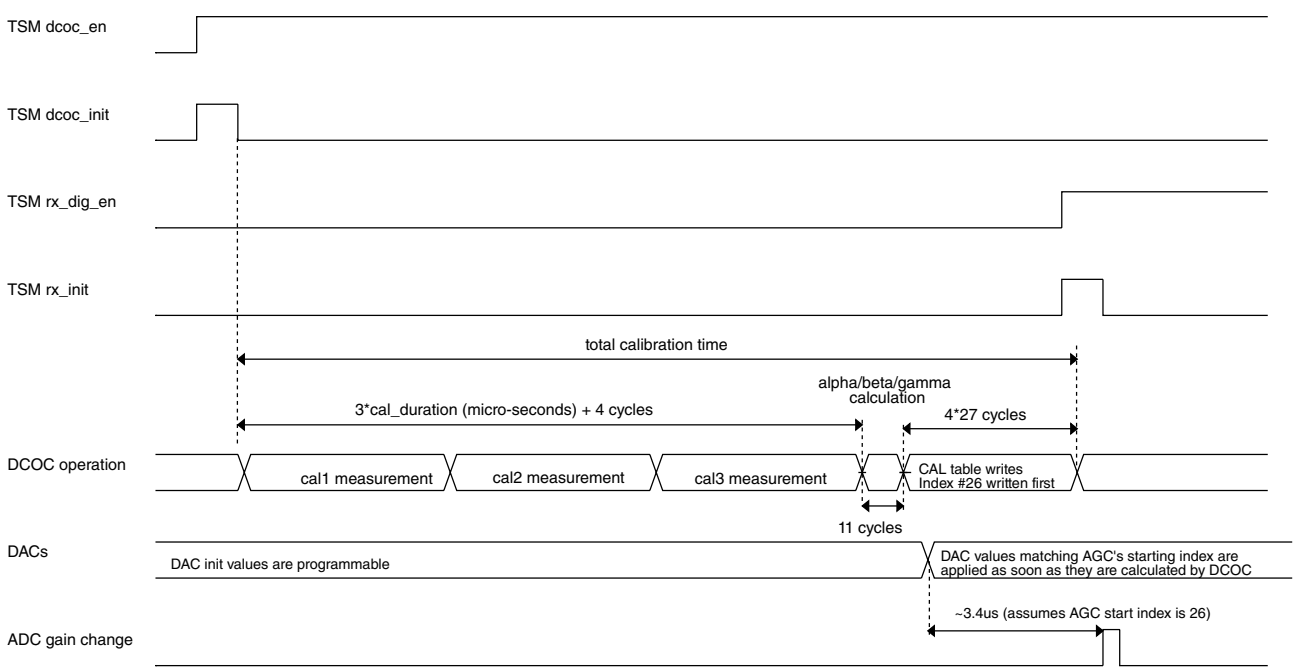


Figure 45-41. DCOC Calibration Timing Diagram

45.3.3.2.5.3.1 DC Offset Estimation during Calibration Phase

During the calibration phase, the DCOC makes DC offset estimates at three distinct RX LNA/BBA gain points. The three gain points are programmed in the 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 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 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 45-42. 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.

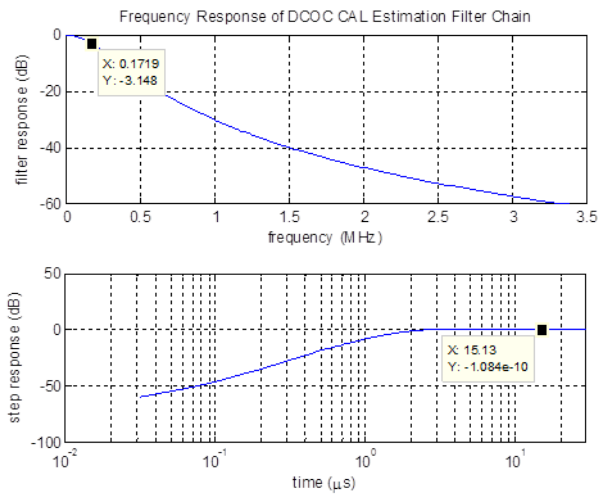


Figure 45-43. Magnitude and Step response of DC Offset Calibration Estimator

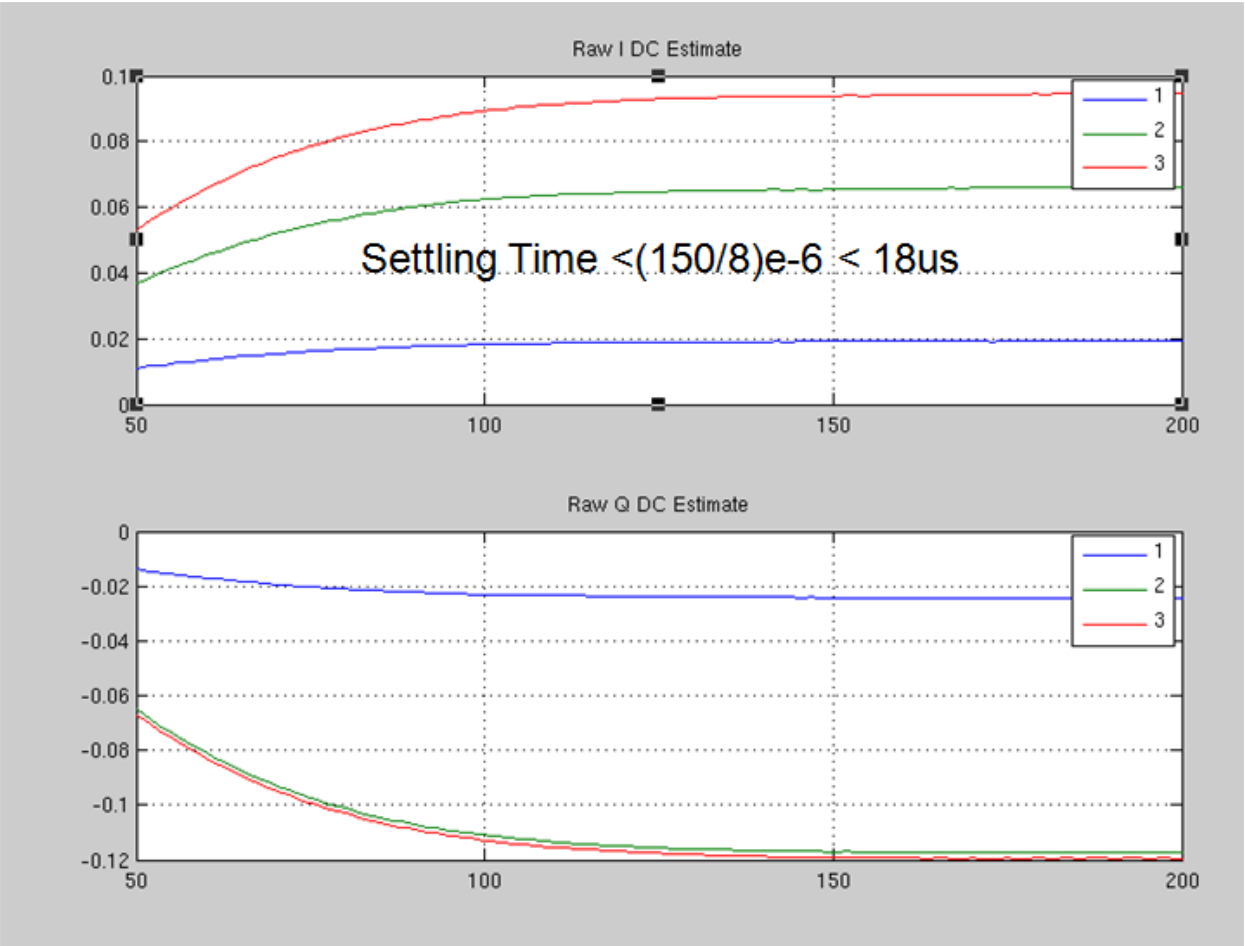


Figure 45-44. 3-step {I,Q} DC estimation during calibration

For debug, the DC offset estimates for the three gain points can be read after calibration from the DCOC_CAL1, DCOC_CAL2, and DCOC_CAL3 registers.

The DCOC operates on the decimated output of the ADC. As BTLE and 802.15.4 have different oversample rates, different IIR filter settings are expected to be used assuming a settling time of 20us is desired. Recommended sets of IIR1/2/3 parameters are shown below.

- Decimation rate 8: $\alpha_1 = 1/8$, $\alpha_2 = 1/16$, $\alpha_3 = 1/16$
- Decimation rate 16: $\alpha_1 = 1/4$, $\alpha_2 = 1/8$, $\alpha_3 = 1/8$

45.3.3.2.5.3.2 DC Estimation at 3 operation gain points of the receiver

The 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 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 LNA and BBA gains
- β - A TZA DC offset which scales with BBA gains
- γ - represents an un-scaled DC offset at the ADC input

These DC components are illustrated in the figure below.

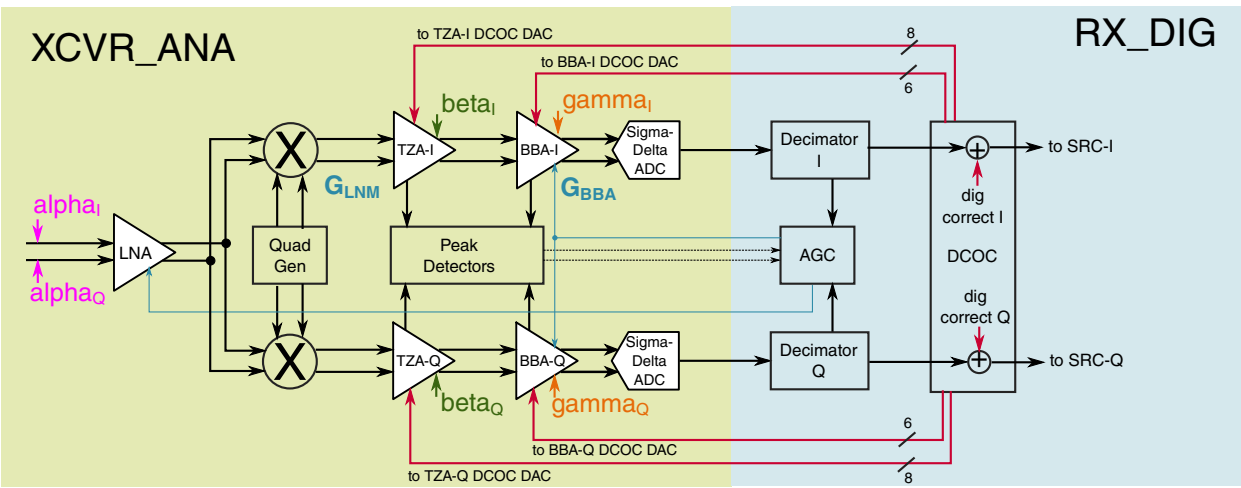


Figure 45-45. Receiver model showing three contributors of DC offset

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 = 1.2V/2^7*(22.9kΩ/34kΩ) = 6.31mV
- the BBA output using a 6-bit DACs with LSB = 1.2V/2^5*(63.44kΩ/120kΩ) = 19.83mV

Calibration is performed open-loop. During calibration, the DCOC normally measures the DC values while the DACs are set to apply an offset of 0V. However, other DAC values can be used during calibration by using the TZA_DCOC_INIT_Q/I and BBA_DCOC_INIT_Q/I bitfields.

NOTE

The equations below are for the case when the TZA_DCOC_INIT_Q/I and BBA_DCOC_INIT_Q/I use their default reset values (which represent DAC values of 0). The α-hat/β-hat/γ-hat calculations in the RTL also support the scenario where non-default values are used; in that case, the α-

hat/β-hat/γ-hat are calculated such they should yield the same estimates as if DAC values of 0 were used.

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

$$RxDCOC\_pre\_ADC \cong (\alpha * G_L + \beta) * G_B + \gamma$$

where

- G_L – Voltage gain of the LNA stage (in linear domain)
- G_B – Voltage gain of the BBA stage (in linear domain)

The ADC has an effective gain of 1code/1.7V. In the decimator, there is a decimation-rate-dependent gain factor, and also a programmable gain factor as described in Decimation Filter. Let ADC_gain represent the combined gain through the ADC and decimator in terms of codes/mV:

```
ADC_gain = 1/1700mV * (OSR^4) * (OSR_scale_factor) * RX_DEC_FILT_GAIN / 2
```

For example, if OSR is 8 (OSR_scale_factor is therefore 2^0 as shown in Table 45-19) and RX_DEC_FILT_GAIN is programmed for a gain of 1.6875, ADC_gain is 2.033codes/mV.

Note that the ADC_gain factor is a component of the value programmed for the BBA_DCOC_STEP, BBA_DCOC_STEP_RECIP, DCOC_TZA_STEP_GAIN, and DCOC_TZA_STEP_RECIP bitfields.

NOTE

The DCOC calibration actually performs slighty better when the ADC_gain factor shown above is multiplied by a scaling factor of ~0.96; this is due to the fixed point arithmetic implementation of the calibration algorithm. The scaling factor effectively causes the calibration algorithm to compute slightly larger TDAC and BDAC values (see below) than it would otherwise calculate.

The total DC offset observed at the RxDCOC is

$$RxDCOC\_post\_ADC \cong ADC\_gain * ((\alpha * G_L + \beta) * G_B + \gamma)$$

The DC offset calibration is carried at three distinct {LNA, BBA} gain settings, as summarized below.

Table 45-29. DC offset calibration at selected {LNA,BBA} gain settings

DCOC_CAL	LNA linear gain (V/V)	BBA 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 DCOC_CAL_GAIN register. Here is an example:

Table 45-30. Example DCOC_CAL_GAIN programming

DCOC_CAL	DCOC LNA CAL GAIN index	DCOC BBA 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]

$$\alpha\text{-hat} = \alpha * ADC\_gain = (DCOC[2] - DCOC[1]) * [1.0((G_{L\_HI}-G_{L\_LO>})*G_{B\_LO})] \quad [4]$$

Using eq. [1] & eq. [3]

$$DCOC\_tmp = (\alpha\text{-hat} * G_{L\_LO} + \beta\text{-hat})*ADC\_gain$$
$$= (DCOC[3] - DCOC[1]) * [1.0/(G_{B\_HI} - G_{B\_LO})] \quad [5]$$

Normally, the TZA_DCOC_INIT bitfield is left at its default value of 0x80 in which case the TZA DCOC DAC applies an offset of 0V. In this case, using eq. [4] & eq. [5]

$$\beta\text{-hat} = \beta * ADC\_gain = [5] - [4]*G_{L\_LO} \quad [6]$$

To account for scenarios where TZA_DCOC_INIT is not 0x80, equation [6] can be generalized as shown below to subtract the contribution from the TZA DCOC DAC. In equation [6a], "tza_dac_sign" is +1 if DCOC_CTRL_0[TZA_CORR_POL] is clear, and -1 if it is set.

$$\beta\text{-hat} = \beta * ADC\_gain = [5] - [4]*G_{L\_LO} - (TZA\_DCOC\_INIT - 0x80)*TZA\_dcoc\_step\_norm*tza\_dac\_sign \quad [6a]$$

Likewise, the BBA_DCOC_INIT bitfield is normally left at its default value of 0x20 in which case the BBA DCOC DAC applies an offset of 0V. In this case using eq. [5] & eq. [3]

$$\gamma\text{-hat} = \gamma * ADC\_gain = [3] - [5]* G_{B\_HI} \quad [7]$$

To account for where scenarios where BBA_DCOC_INIT is not 0x20, equation [7] can be generalized as shown below to subtract the contribution from the BBA DCOC DAC. In equation [7a], "bba_dac_sign" is +1 if DCOC_CTRL_0[BBA_CORR_POL] is clear, and -1 if it is set.

$$\gamma\text{-hat} = \gamma * \text{ADC\_gain} = [3] - [5] * G_{B\_HI} - (\text{BBA\_DCOC\_INIT} - 0x20) * \text{BBA\_dcoc\_step} * \text{bba\_dac\_sign} \quad [7a]$$

After calibration, α-hat, β-hat, and γ-hat can be read from the DCOC_CAL_ALPHA, DCOC_CAL_BETA, and 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:

- 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.
- DCOC_CAL_RCP[DCOC_TMP_CALC_RECIP]. This is the $1.0/(G_{B\_HI}- G_{B\_LO})$ factor from equation [5] above.
- LNA_GAIN_LIN_VAL_x (LNA_GAIN_LIN_VAL_x_y registers). The G_{L_LO} and G_{L_HI} gains are found from the appropriate index into this table.
- BBA_RES_TUNE_LIN_VAL_x (BBA_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.
- DCOC_CTRL_0[TZA_CORR_POL] bit determines the polarity of the TZA DCOC DAC value applied during calibration.
- DCOC_CTRL_0[BBA_CORR_POL] bit determines the polarity of the BBA DCOC INIT DAC value applied during calibration.

45.3.3.2.5.3.3 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 (AGC_GAIN_TBL_xx_yy registers) can be calculated and stored (in the DCOC_OFFSET_n registers).

If DCOC_CTRL_0[DCOC_CAL_CHECK_EN] is set, the α-hat and β-hat estimates are checked against the thresholds in the DCOC_CAL_FAIL_TH and DCOC_CAL_PASS_TH registers as a condition of updating the DCOC_OFFSET_n registers:

- If the absolute value of all of the α-hat and β-hat estimates are less than the "pass" thresholds, the DCOC_OFFSET_n tables are updated.

- If the absolute value of any of the α-hat and β-hat estimates are greater than the "fail" thresholds, the DCOC_OFFSET_n tables are not updated.
- If neither a "pass" nor a "fail" condition is detected, the DCOC_OFFSET_n tables are only updated if a "pass" condition has not occurred since the last reset

In the cases where the DCOC_OFFSET_n table is not updated, the DCOC will use whatever values were previously written into the table to perform corrections during the burst.

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=26, 25, ... 0 entries of the AGC_GAIN_TBL, and where $TDAC_k$ and $BDAC_k$ represent the TZA and BBA DAC values:

```
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_BBA_TOTAL = DCOC_TZA_residual *G_Bk + γ-hat
BDAC_k = round(DCOC_BBA_TOTAL*[1.0/BBA_dcoc_step])
```

Information derived from the programmable AGC_GAIN_TBL_xx_yy registers		Computed and stored in the DCOC_OFFSET_n registers			
LNA linear gain (V/V)	BBA linear gain (V/V)	TZA-I DAC code	BBA-I DAC Code	TZA-Q DAC code	BBA-Q DAC Code
G_{L26}	G_{B26}	$TDAC_{I26}$	$BDAC_{I26}$	$TDAC_{Q26}$	$BDAC_{Q26}$
…	…	…	…	…	…
G_{L0}	G_{B0}	$TDAC_{I0}$	$BDAC_{I0}$	$TDAC_{Q0}$	$BDAC_{Q0}$

Note that the values stored in the DCOC_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 DCOC_CTRL_0[TZA_CORR_POL] and DCOC_CTRL_0[BBA_CORR_POL] bits. Also, note that the DAC code values themselves (which are reflected in the 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 DCOC_OFFSET_n registers, the DCOC uses the following register bitfields:

- DCOC_BBA_STEP[BBA_DCOC_STEP_RECIP]. This is the 1.0/BBA_dcoc_step referred to above.
- DCOC_TZA_STEP_0[DCOC_TZA_STEP_RCP]. This is the 1.0/ TZA_dcoc_step_norm referred to above.
- DCOC_TZA_STEP_0[DCOC_TZA_STEP]. This is the TZA_dcoc_step_norm referred to above.

- LNA_GAIN_LIN_VAL_x (LNA_GAIN_LIN_VAL_x_y registers). This is used for the G_{Lk} referred to above
- BBA_RES_TUNE_LIN_VAL_x (BBA_RES_TUNE_LIN_VAL_x_y registers). This is used for the G_{Bk} referred to above
- RF_TZA[RX_TZA_SPARE]. The RX_TZA_SPARE[2] bit specifies the allowed range of values for the TZA DCOC DAC.
- DCOC_CTRL_0[DCOC_CAL_CHECK_EN]. Determines whether calibration check logic is enabled or not. If it is, then the pass and fail threshold registers are used:
  - DCOC_CAL_PASS_TH[DCOC_CAL_ALPHA_P_TH]. Alpha pass threshold
  - DCOC_CAL_PASS_TH[DCOC_CAL_BETA_P_TH]. Beta pass threshold
  - DCOC_CAL_FAIL_TH[DCOC_CAL_ALPHA_F_TH]. Alpha fail threshold
  - DCOC_CAL_FAIL_TH[DCOC_CAL_BETA_F_TH]. Beta fail threshold

45.3.3.2.5.3.4 Use of DC offset Table After Calibration

After calibration, when the TSM RX_DIG_EN is asserted, if the DCOC_CTRL_0[DCOC_CORRECT_EN] bit is 1, the DCOC will apply the value stored in the DCOC_OFFSET_n table on each AGC gain change. As indicated previously, the value stored in the DCOC_OFFSET_n register represents the DAC vaue of the DC offset.

If the DCOC_CTRL_0[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 DCOC_CTRL_0[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 DCOC_OFFSET_n table. In this case, the RX_DIG_CTRL[RX_DCOC_CAL_EN] bit should be cleared so that the DCOC will not overwrite the DCOC_OFFSET_n table, but DCOC_CTRL_0[DCOC_CORRECT_EN] bit should still be set.

45.3.3.2.5.4 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 DCOC_CTRL_0[DCOC_CORRECT_SRC] bit is programmed to 1 (RX_DIG_CTRL[RX_DCOC_EN] and DCOC_CTRL_0[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, and resets the digital correction to zero.
- If the AGC gain table index is larger than or equal to DCOC_CTRL_1[DCOC_TRK_MIN_AGC_IDX], then...
  - After the AGC gain settles (refer to AGC_CTRL_1[LNA_GAIN_SETTLE_TIME] and AGC_CTRL_2[BBA_GAIN_SETTLE_TIME] bitfields) , the tracker's estimate is reset to either 0 or the current I/Q sample, depending on programming of the DCOC_CTRL_0[TRACK_FROM_ZERO] bit.
  - The DCOC will periodically adjust the DACs and digital correction at intervals defined by the DCOC_CTRL_0[DCOC_CORR_DLY] bitfield until a timeout defined by the 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 DCOC_CTRL_1[DCOC_TRK_EST_GS_CNT].
  - If the DCOC_CTRL_0[DCOC_MIDPWR_TRK_DIS] bit is set, the tracking corrections are disabled if either the TZA or BBA lo peak detector asserts. This is implemented by resetting the counters associated with DCOC_CORR_DLY and DCOC_HOLD_TIME, so if the lo peak deters stop asserting, then tracking corrections woud be resumed. The counter associated with DCOC_TRK_EST_GS_CNT is not affected by the DCOC_MIDPWR_TRK_DIS bit.
- If the AGC gain table index is smaller than DCOC_CTRL_1[DCOC_TRK_MIN_AGC_IDX], then tracking is not activated for this gain setting.

The figure below illustrates the timing behavior during DC offset tracking correction after an AGC gain change. If the 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 DCOC_TRK_EST_GS_CNT=2, so the tracking estimator changes its parameters after 2 update correction periods.

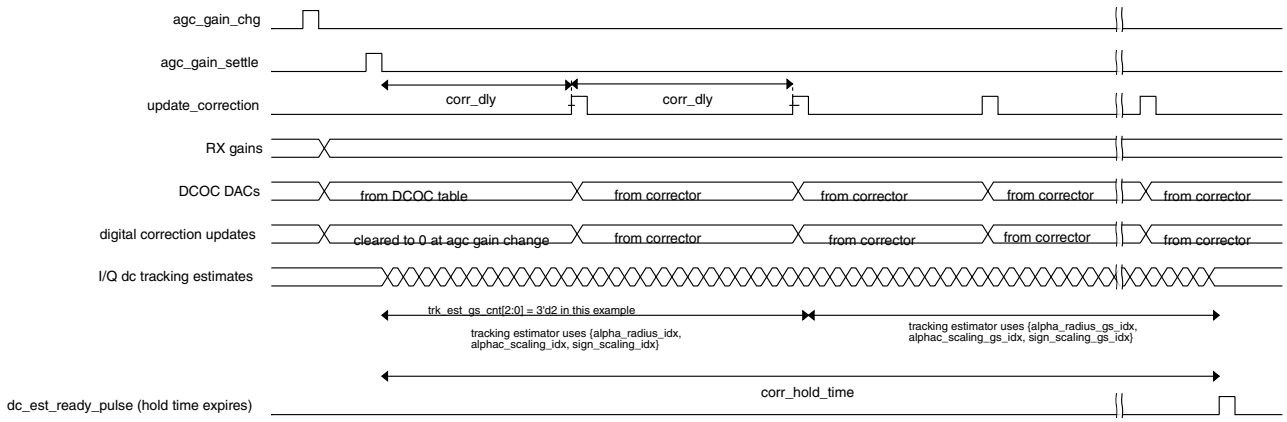


Figure 45-46. DC Offset Tracking Correction Timing

45.3.3.2.5.4.1 DC Offset Estimation during Tracking Phase

There are 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, alphac_scale, and sign_scale parameters, 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_gs_idx} after the number of update corrections defined by DCOC_TRK_EST_GS_CNT.

- DCOC_CTRL_1[DCOC_ALPHA_RADIUS_IDX] and DCOC_CTRL_1[DCOC_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".
- DCOC_CTRL_1[DCOC_ALPHAC_SCALE_IDX] and DCOC_CTRL_1[DCOC_ALPHAC_SCALE_GS_IDX]. Step size for I/Q center, assumes values of {1/2, 1/4, 1/8, 1/16, 1/32 or 1/64}. In pseudo-code below, this value times the "alpha_radius" value is referred to as "alpha_center"
- DCOC_CTRL_1[DCOC_SIGN_SCALE_IDX] and DCOC_CTRL_1[DCOC_SIGN_SCALE_GS_IDX]. sign_scaling. Scaling factor for sign() based correction of the I/Q center, assumes values of {1/8, 1/16, 1/32, or 1/64}. In psuedo-code below, this is referred to as "sign_scaling".

- DCOC_CTRL_1[DCOC_TRK_EST_GS_CNT]. If this is 0, then the algorithm only uses the {alpha_radius_idx, alphac_scale_idx, sign_scale_idx} parameters above. Otherwise, the algorithm begins with the {alpha_radius_idx, alphac_scale_idx, sign_scale_idx} parameters after each AGC gain change, but then switches to the 2nd parameter set {alpha_radius_gs_idx, alphac_scale_gs_idx, sign_scale_gs_idx} after DCOC_TRK_EST_GS_CNT number of update correction periods.
- DCOC_CTRL_0[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 DCOC_DC_EST register.

The DC correction applied by the tracking algorithm makes use of the following register bitfields:

- DCOC_BBA_STEP[BBA_DCOC_STEP].
- DCOC_BBA_STEP[BBA_DCOC_STEP_RECIP].
- DCOC_TZA_STEPn[TZA_DCOC_STEP], where n=0..10.
- DCOC_TZA_STEPn[TZA_DCOC_STEP_RECIP], where n=0..10.
- DCOC_CTRL_0[TZA_CORR_POL]
- DCOC_CTRL_0[BBA_CORR_POL]
- RF_TZA[RX_TZA_SPARE]. The RX_TZA_SPARE[2] bit specifies the allowed range of values for the TZA DCOC DAC.

Normally, the tracking estimator is only enabled when it is needed by the corrector. However, by setting the DCOC_CTRL_0[TRK_EST_OVR] bit, the tracking estimator can be enabled continuously while the DCOC is active.

45.3.3.2.5.5 DC Offset Manual Mode

For debug purposes, the DCOC supports a manual mode. This mode is enabled when the DCOC_CTRL_0[DCOC_MAN] bit is set, which overrides all other DCOC programmable control bits. In this mode, the DCOC DAC values are applied from DCOC_DAC_INIT[TZA_DCOC_INIT_Q], DCOC_DAC_INIT[TZA_DCOC_INIT_I], DCOC_DAC_INIT[BBA_DCOC_INIT_Q], and DCOC_DAC_INIT[BBA_DCOC_INIT_Q], and the digital correction is applied from DCOC_DIG_MAN[DIG_DCOC_INIT_Q] and DCOC_DIG_MAN[DIG_DCOC_INIT_I].

45.3.3.2.6 Channel Filter

The channel filter is a programmable, 24 tap symmetric FIR filter with 6-bit to 10-bit coefficients. It operates at the decimated sample rate on the output from the DCOC. The coefficient values are programmable and should be configured for the mode of operation (BTLE, 802.15.4, Generic LL). The filter output is scaled by 1/256; the sum of the 12 coefficients programmed is expected to be <=256 in order to avoid saturation. A few example coefficient filter sets are given in the table below.

Table 45-31. Channel Filter Coefficients for 32MHz XO

Coefficient Register	Format	BTLE	802.15.4
RX_CHF_COEF0	s5	4	3
RX_CHF_COEF1	s5	5	10
RX_CHF_COEF2	s6	2	-16
RX_CHF_COEF3	s6	-6	-9
RX_CHF_COEF4	s6	-17	37
RX_CHF_COEF5	s6	-26	-12
RX_CHF_COEF6	s7	-25	-57
RX_CHF_COEF7	s7	-8	61
RX_CHF_COEF8	s8	24	63
RX_CHF_COEF9	s8	66	-156
RX_CHF_COEF10	s9	105	-54
RX_CHF_COEF11	s9	129	385

45.3.3.2.7 Sample Rate Conversion (SRC) Filter

The sample rate conversion block is used for a fractional clock rate conversion of the sampled decimated signal, when 26MHz reference clock is used. In this mode, the ADC is clocked at 2xFref or 52 MHz. For the 2.4GHz targeted modulations, the PHY data rate is always 8x the symbol rate (of 1Mbps, 500kbps or 250kbps) for (G)FSK modes and 4x

the 2Mchips/sec rate for IEEE 802.15.4. These desired frequencies are not related to the ADC clock rate by an integer factor, which necessitates the need for a fractional sample rate conversion that is performed by SRC. For a complete list of the fractional conversion modes refer to Table 45-18 . The SRC is enabled by setting RX_DIG_CTRL[RX_SRC_EN] to 1, and the conversion factor is selected by RX_DIG_CTRL[RX_SRC_RATE] as described below.

45.3.3.2.7.1 Sample Rate Conversion by a factor of 8/13

If RX_DIG_CTRL[RX_SRC_RATE] = 0, the SRC block is configured to perform a data rate conversion by a factor of 8/13, i.e., for every 13 input samples, the SRC block generates 8 output samples at a reduced data rate. A linearly interpolated first order hold (FOH) resampler is used for implementation as it provides significant better anti-aliasing as compared to a zero order hold (ZOH).

45.3.3.2.7.2 Sample Rate Conversion by a factor of 12/13

If RX_DIG_CTRL[RX_SRC_RATE] = 1, the SRC block is configuration a zero-order hold sample rate conversion is used. This mode is realized by skipping one out of every 13 input samples.

45.3.3.2.8 DC Residual Correction

This block is used to compensate for the DC offset which is still present after AGC and DCOC calibration/tracking has completed. It estimates the DC offset by measuring the minimum and maximum value and then applies an additive correction factor to the I and Q channels. This block does not change the value of the DCOC DACs.

The DC offset is compensated independently on I and Q channels. The correction loop determines the minimum and maximum values over a specified window of samples (*Nwin*) which are then averaged and used (with an update parameter *Alpha*) to create the DC estimate. This DC estimate is then subtracted from the input samples. The applied DC estimate continues to be updated every *NWIN* samples until *IT_FREEZE* of these windows have elapsed, at which time the DC estimate is frozen.

Multiple modes of operations can be identified for residual DC offset compensation block. They are presented in table below.

Table 45-32. Residual DC offset compensation operational modes

Mode	Description	DC_RESID_EN	EXT_DC_EN	IT_FREEZE	DLY	NWIN, ALPHA

Table continues on the next page...

Table 45-32. Residual DC offset compensation operational modes (continued)

Bypass	The DC residual is disabled. The input I and Q samples to the DC Residual block are passed directly to the DC Redisdual output without any modifications. .	0	-	-	-	-
Mode 1	In this mode, DC residual has the role of calculating and applying a DC offset without use of an initial estimate from the DCOC. The DC residual processing is restarted at every AGC gain change This mode is suitable for use when an AGC gain change occurs which is not followed by DCOC tracking.	1	0	>1	>0	program as needed
Mode 2	In this mode, DC residual has the role of refining the DC offset estimate provided by the DCOC at the end of the DCOC tracking after the last DCOC DAC update. The DC residual processing is restarted whenever DCOC's tracking hold timer expires. This mode is suitable after AGC gain change followed by DCOC tracking.	1	1	>1	-	program as needed
Mode 3	In this mode, DC residual has the role of applying a correction based on the DC offset estimate provided by the DCOC at the end of the DCOC tracking after the last DCOC DAC update. The DC residual processing is restarted whenever DCOC's tracking hold timer expires. This mode is suitable after AGC gain change and DCOC tracking when no further refining is desired.	1	1	1	-	-

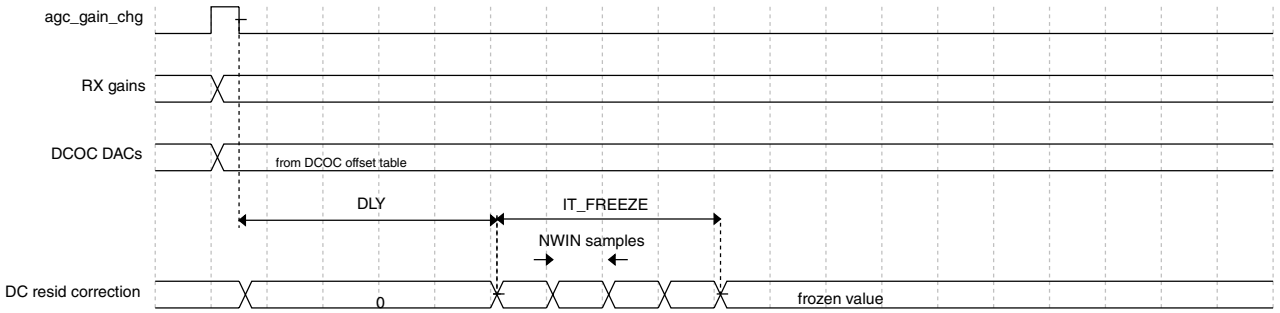


Figure 45-47. DC residual mode 1 timing

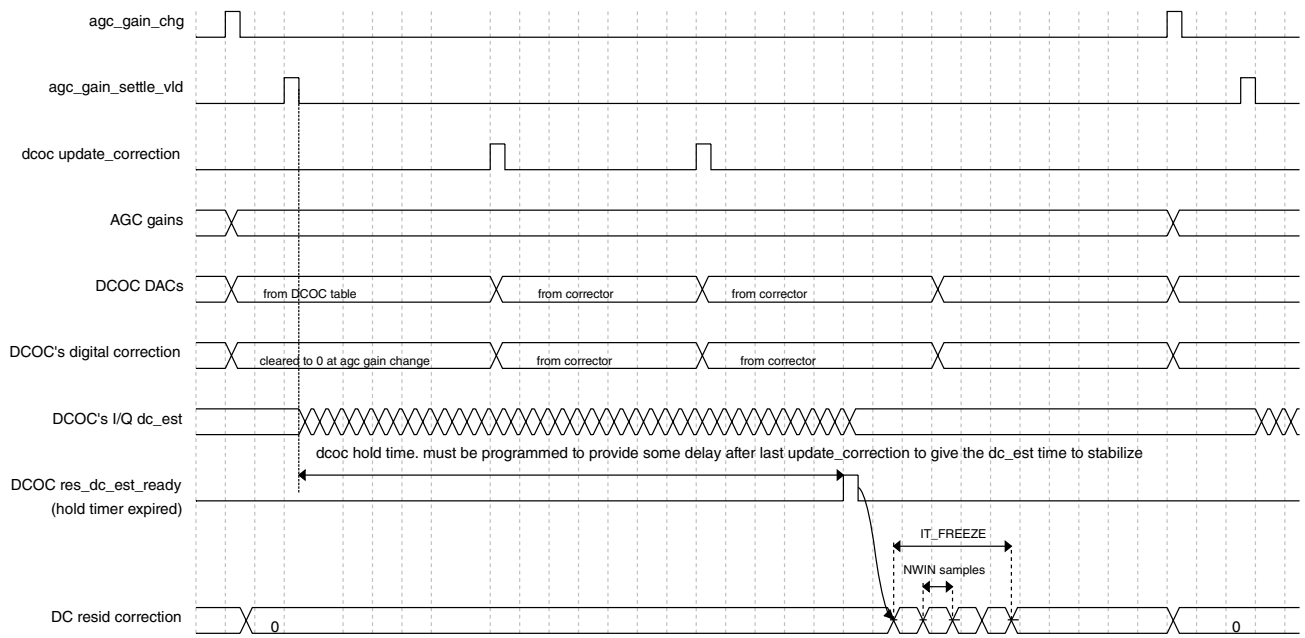


Figure 45-48. DC residual mode 2 timing

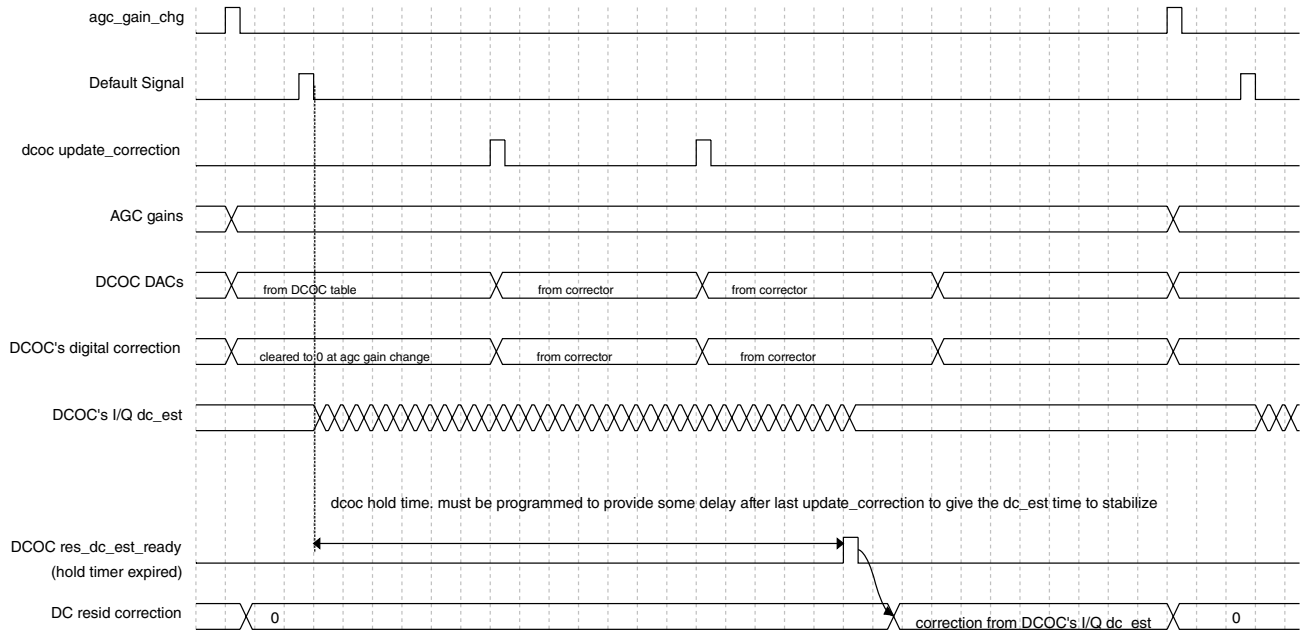


Figure 45-49. DC residual mode 3 timing

The behavior of the Resdual DC block is also dependent on the AGC index and the values programmed for DC_RESID's MIN_AGC_IDX and the DCOC's DCOC_CTRL_1[DCOC_TRK_MIN_AGC_IDX]. This behavior is defined in the table below, which assumes DC_RESID_EN=1.

Table 45-33. Residual DC offset compensation operation dependency on AGC table index

EXT_DC_EN	MIN_AGC_IDX	DCOC_TRK_MIN_AGC_IDX	Behavior
-	AGC_IDX < MIN_AGC_IDX	-	DC residual is disabled
0	AGC_IDX >= MIN_AGC_IDX	-	DC residual operates in Mode 1 (it does not use an initial estimate from the DCOC)
1		AGC_IDX >= DCOC_TRK_MIN_AGC_IDX	DC residual operates in Mode 2 or 3 (It uses the estimate from the DCOC)
		AGC_IDX < DCOC_TRK_MIN_AGC_IDX	DC residual operates in Mode 1. it does not use the estimate from the DCOC, since the DCOC's tracker is disabled for this AGC gain index

45.3.3.2.9 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.

NOTE

The normalizer's 5-bit I/Q output is currently not used.

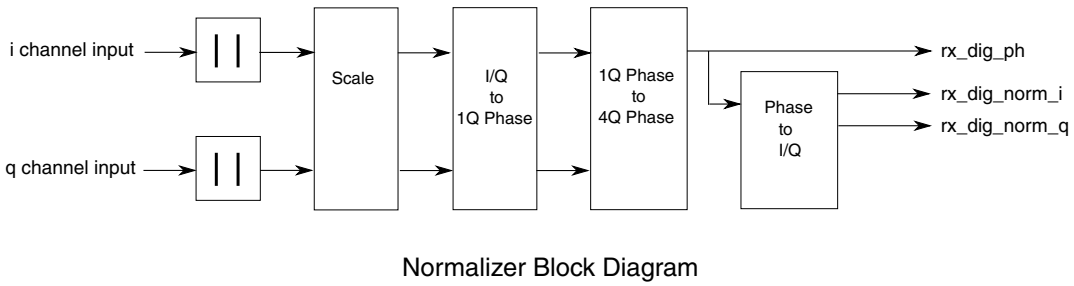


Figure 45-50. Normalizer Block Diagram

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.

45.3.3.2.10 AuxPLL Frequency Calibration

The AuxPLL Frequency Calibration module finds the best calibration code associated with the AuxPLL. Calibration is initiated using a TSM control signal, and it is expected that this calibration will be performed on each RX warmup. Features of the module include the following:

- Supports two different calibration run window lengths: 64 reference clock cycles, and 128 reference clock cycles. For a 32Mhz reference clock, the overall calibration time is <17us for a run window of 64 reference clock cycles, and <31us for a run window of 128 reference clock cycles.
- Calibration logic can be disabled and a manual calibration value can be applied
- The clk cycle at which the fcal_count[9:0] input from the analog is sampled is programmable.

The module uses a binary search to find the best 7bit calibration code (auxpll_fco_dac_cal_adjust[6:0]). At the end of each calibration stage, the auxpll_fcal_count[9:0] value output from the analog (which counts the number of positive and negative edges of the auxpll clock during the run window) is sampled and compared against the expected count (256 for a run window length of 64 clk cycles, 512

for a run window length of 128 clk cycles), and a decision is made whether the appropriate auxpll_vco_dac_cal_adjust[6:0] bit should be set or cleared. The decision logic is programmable:

- AUXPLL_FCAL_CTRL[FCAL_COMP_INV]=1'b0 (default behavior): if the fcal_count is larger than or equal to the expected count, the cal_adjust bit remains set.
- AUXPLL_FCAL_CTRL[FCAL_COMP_INV]=1'b1: if the fcal_count is less than or equal to the expected count, the cal_adjust bit remains set.

After each of the seven calibration cycles, the current count error is compared to the best (smallest absolute) count error found so far, and after the binary search is complete, the module outputs the calibration code associated with the smallest absolute count error which was found.

Debug features include the following:

- Count value for each of the several calibration stages are saved to registers
- The best absolute count error found during calibration is saved to a register
- Calibration logic can be disabled and a manual calibration value can be applied

The figure below shows the beginning of the AuxPLL calibration timing. For a run window length of 64 clk cycles (which is the configuration expected to be used) each of the 7 calibration stages requires ~76clks, with total calibration time of ~532 clks, which is <17us at 32MHz. Only the 1st (and the beinning of the 2nd) of the 7 calibration stages are shown in the figure.

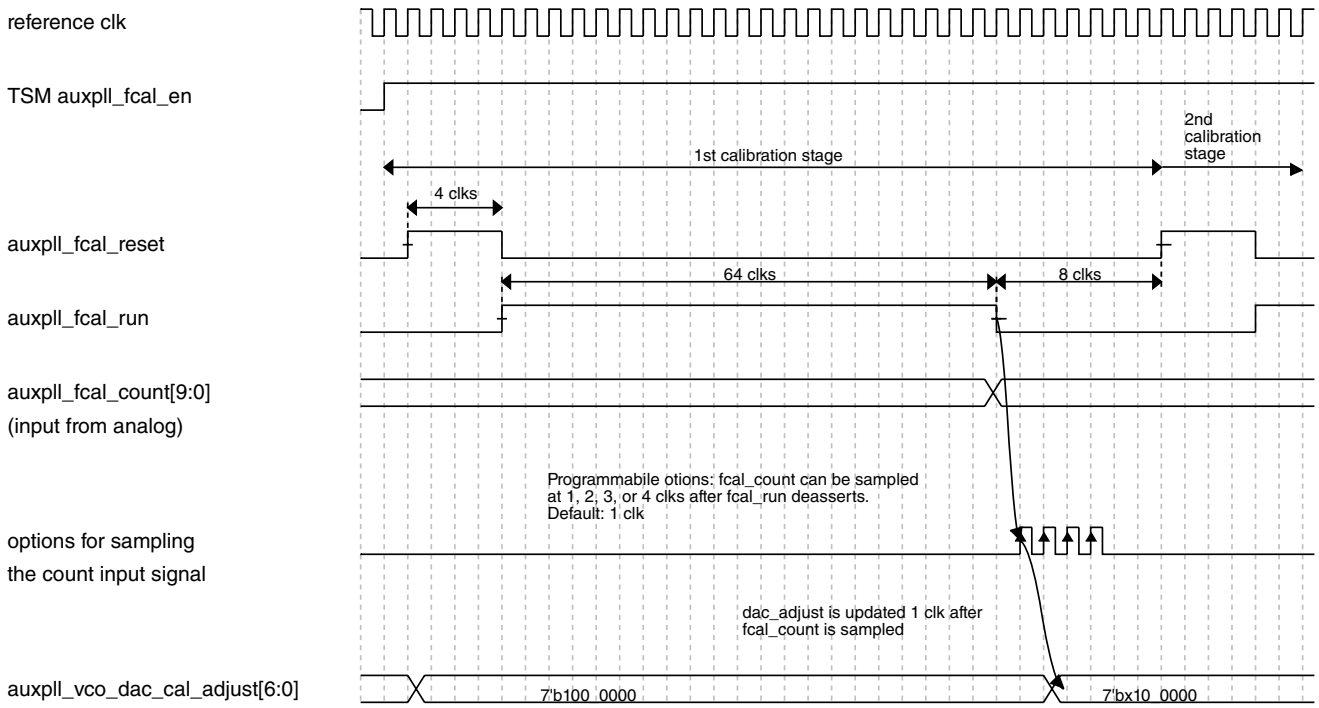


Figure 45-51. RX AuxPLL FCAL timing

45.3.3.2.11 RX RC Calibration

The RX RC Calibration module finds the best 5bit calibration code for the RC calibration circuit in the analog. From the resulting RC calibration code, separate 5bit codes are derived for each of the following analog circuits: TZA, BBA, BBA2, and ADC. The calibration sequence is initiated by a TSM control signal, and it is expected that this calibration will be performed on each RX warmup.

Features of the module include the following:

- The run window length for each of the five calibration stages is fixed at 4us. The total calibration time is 1us + 5*(4us + 10clks + ipr_rccal_smpl_dly[1:0] clks) - 1 clk.
  - For a 32MHz clk, this is <=23us for all values of ipr_rccal_smpl_dly[1:0]
  - For a 26MHz, this is <=23us for ipr_rccal_smpl_dly[1:0] = 2'b0, and <=24us for other values.
- The four (TZA, BBA, BBA2, and ADC) calibration codes are derived from the RC calibration code by adding signed 4bit programmable offsets. Each of TZA, BBA, BBA2 and ADC have their own programmable offset with saturation detection.
- The comparator output (rccal_comp_out in figure below) from the analog can be programmably inverted.

- The clk cycle at which the comp_out input from the analog is sampled is programmable.
- The output for each of the individual calibration codes (TZA, BBA, BBA2, and ADC) can be applied manually from a register

The module uses a binary search to find the best 5bit calibration code (rccal_code[4:0]). At the end of each calibration stage, the comp_out value output from the analog is sampled, optionally inverted, and used to determine whether the appropriate rccal_code[4:0] bit should be set or cleared.

The figure below illustrates the RC calibration timing. There is a 1us initialization after the calibration circuit is enabled, and then each of the 5 calibration stages requires 5us, for a total of 26us for the entire calibration sequence. Only the inialization 1us and the 1st calibration stage are shown in the figure. The rccal_comp_out signal level at end the end of the calibration stage is used to determine whether the appropriate bit of rccal_code[4:0] will be set or cleared.

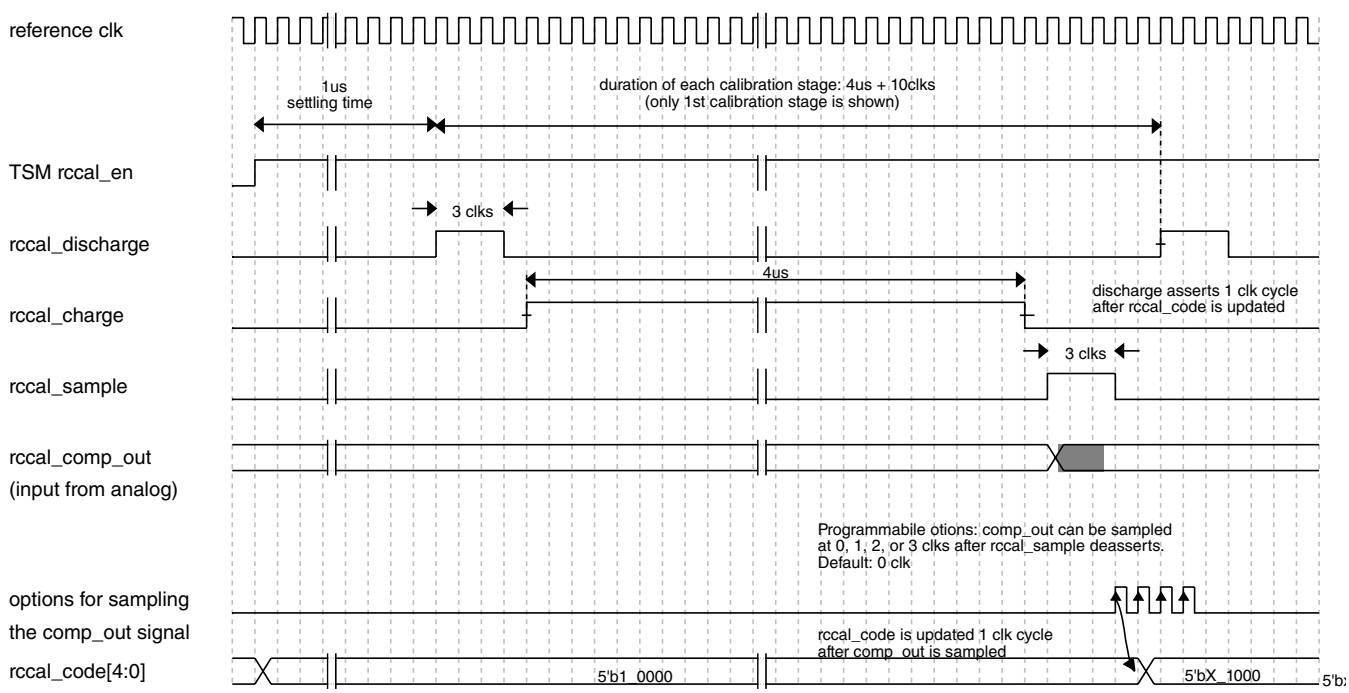


Figure 45-52. RX RCCAL timing

45.3.3.3 Memory Map and register definition

The rx_dig memory map and description of the rx_dig registers is included in the following register section.

45.3.3.3.1 XCVR_RX_DIG register descriptions

45.3.3.3.1.1 XCVR_RX_DIG_ADDR Memory map

XCVR_RX_DIG base address: 4005_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	RX Digital Control (RX_DIG_CTRL)	32	RW	0000_0000h
4h	AGC Control 0 (AGC_CTRL_0)	32	RW	0000_0000h
8h	AGC Control 1 (AGC_CTRL_1)	32	RW	0000_0000h
Ch	AGC Control 2 (AGC_CTRL_2)	32	RW	00A6_9000h
10h	AGC Control 3 (AGC_CTRL_3)	32	RW	0000_0000h
14h	AGC Status (AGC_STAT)	32	RO	0000_0000h
18h	RSSI Control 0 (RSSI_CTRL_0)	32	RW	0030_0000h
1Ch	RSSI Control 1 (RSSI_CTRL_1)	32	RO	0000_0000h
24h	DCOC Control 0 (DCOC_CTRL_0)	32	RW	0000_0000h
28h	DCOC Control 1 (DCOC_CTRL_1)	32	RW	0000_0000h
2Ch	DCOC DAC Initialization (DCOC_DAC_INIT)	32	RW	8080_2020h
30h	DCOC Digital Correction Manual Override (DCOC_DIG_MAN)	32	RW	0000_0000h
34h	DCOC Calibration Gain (DCOC_CAL_GAIN)	32	RW	0000_0000h
38h	DCOC Status (DCOC_STAT)	32	RO	8080_2020h
3Ch	DCOC DC Estimate (DCOC_DC_EST)	32	RO	0000_0000h
40h	DCOC Calibration Reciprocals (DCOC_CAL_RCP)	32	RW	0000_0000h
48h	IQMC Control (IQMC_CTRL)	32	RW	0400_8000h
4Ch	IQMC Calibration (IQMC_CAL)	32	RW	0000_0400h
50h	LNA_GAIN Step Values 3..0 (LNA_GAIN_VAL_3_0)	32	RW	3809_321Dh
54h	LNA_GAIN Step Values 7..4 (LNA_GAIN_VAL_7_4)	32	RW	8B74_5D4Fh
58h	LNA_GAIN Step Values 8 (LNA_GAIN_VAL_8)	32	RW	0000_B6A1h
5Ch	BBA Resistor Tune Values 7..0 (BBA_RES_TUNE_VAL_7_0)	32	RW	0000_0000h
60h	BBA Resistor Tune Values 10..8 (BBA_RES_TUNE_VAL_10_8)	32	RW	0000_0000h
64h	LNA Linear Gain Values 2..0 (LNA_GAIN_LIN_VAL_2_0)	32	RW	0000_0000h
68h	LNA Linear Gain Values 5..3 (LNA_GAIN_LIN_VAL_5_3)	32	RW	0000_0000h
6Ch	LNA Linear Gain Values 8..6 (LNA_GAIN_LIN_VAL_8_6)	32	RW	0000_0000h
70h	LNA Linear Gain Values 9 (LNA_GAIN_LIN_VAL_9)	32	RW	0000_0000h
74h	BBA Resistor Tune Values 3..0 (BBA_RES_TUNE_LIN_VAL_3_0)	32	RW	0000_0000h
78h	BBA Resistor Tune Values 7..4 (BBA_RES_TUNE_LIN_VAL_7_4)	32	RW	0000_0000h
7Ch	BBA Resistor Tune Values 10..8 (BBA_RES_TUNE_LIN_VAL_10_8)	32	RW	0000_0000h
80h	AGC Gain Tables Step 03..00 (AGC_GAIN_TBL_03_00)	32	RW	0000_0000h
84h	AGC Gain Tables Step 07..04 (AGC_GAIN_TBL_07_04)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
88h	AGC Gain Tables Step 11..08 (AGC_GAIN_TBL_11_08)	32	RW	0000_0000h
8Ch	AGC Gain Tables Step 15..12 (AGC_GAIN_TBL_15_12)	32	RW	0000_0000h
90h	AGC Gain Tables Step 19..16 (AGC_GAIN_TBL_19_16)	32	RW	0000_0000h
94h	AGC Gain Tables Step 23..20 (AGC_GAIN_TBL_23_20)	32	RW	0000_0000h
98h	AGC Gain Tables Step 26..24 (AGC_GAIN_TBL_26_24)	32	RW	0000_0000h
A0h - 108h	DCOC Offset (DCOC_OFFSET_0 - DCOC_OFFSET_26)	32	RW	0000_0000h
10Ch	DCOC BBA DAC Step (DCOC_BBA_STEP)	32	RW	0000_0000h
110h	DCOC TZA DAC Step 0 (DCOC_TZA_STEP_0)	32	RW	0000_0000h
114h	DCOC TZA DAC Step 1 (DCOC_TZA_STEP_1)	32	RW	0000_0000h
118h	DCOC TZA DAC Step 2 (DCOC_TZA_STEP_2)	32	RW	0000_0000h
11Ch	DCOC TZA DAC Step 3 (DCOC_TZA_STEP_3)	32	RW	0000_0000h
120h	DCOC TZA DAC Step 4 (DCOC_TZA_STEP_4)	32	RW	0000_0000h
124h	DCOC TZA DAC Step 5 (DCOC_TZA_STEP_5)	32	RW	0000_0000h
128h	DCOC TZA DAC Step 6 (DCOC_TZA_STEP_6)	32	RW	0000_0000h
12Ch	DCOC TZA DAC Step 7 (DCOC_TZA_STEP_7)	32	RW	0000_0000h
130h	DCOC TZA DAC Step 5 (DCOC_TZA_STEP_8)	32	RW	0000_0000h
134h	DCOC TZA DAC Step 9 (DCOC_TZA_STEP_9)	32	RW	0000_0000h
138h	DCOC TZA DAC Step 10 (DCOC_TZA_STEP_10)	32	RW	0000_0000h
160h	DCOC Calibration Fail Thresholds (DCOC_CAL_FAIL_TH)	32	RW	0000_0000h
164h	DCOC Calibration Pass Thresholds (DCOC_CAL_PASS_TH)	32	RW	0000_0000h
168h	DCOC Calibration Alpha (DCOC_CAL_ALPHA)	32	RO	0000_0000h
16Ch	DCOC Calibration Beta Q (DCOC_CAL_BETA_Q)	32	RO	0000_0000h
170h	DCOC Calibration Beta I (DCOC_CAL_BETA_I)	32	RO	0000_0000h
174h	DCOC Calibration Gamma (DCOC_CAL_GAMMA)	32	RO	0000_0000h
178h	DCOC Calibration IIR (DCOC_CAL_IIR)	32	RW	0000_0000h
180h - 188h	DCOC Calibration Result (DCOC_CAL1 - DCOC_CAL3)	32	RO	0000_0000h
190h	RX_DIG CCA ED LQI Control Register 0 (CCA_ED_LQI_CTRL_0)	32	RW	0000_0000h
194h	RX_DIG CCA ED LQI Control Register 1 (CCA_ED_LQI_CTRL_1)	32	RW	0000_0000h
198h	RX_DIG CCA ED LQI Status Register 0 (CCA_ED_LQI_STAT_0)	32	RO	0000_0000h
1A0h	Receive Channel Filter Coefficient 0 (RX_CHF_COEF_0)	32	RW	0000_0000h
1A4h	Receive Channel Filter Coefficient 1 (RX_CHF_COEF_1)	32	RW	0000_0000h
1A8h	Receive Channel Filter Coefficient 2 (RX_CHF_COEF_2)	32	RW	0000_0000h
1ACh	Receive Channel Filter Coefficient 3 (RX_CHF_COEF_3)	32	RW	0000_0000h
1B0h	Receive Channel Filter Coefficient 4 (RX_CHF_COEF_4)	32	RW	0000_0000h
1B4h	Receive Channel Filter Coefficient 5 (RX_CHF_COEF_5)	32	RW	0000_0000h
1B8h	Receive Channel Filter Coefficient 6 (RX_CHF_COEF_6)	32	RW	0000_0000h
1BCh	Receive Channel Filter Coefficient 7 (RX_CHF_COEF_7)	32	RW	0000_0000h
1C0h	Receive Channel Filter Coefficient 8 (RX_CHF_COEF_8)	32	RW	0000_0000h
1C4h	Receive Channel Filter Coefficient 9 (RX_CHF_COEF_9)	32	RW	0000_0000h
1C8h	Receive Channel Filter Coefficient 10 (RX_CHF_COEF_10)	32	RW	0000_0000h

Table continues on the next page...

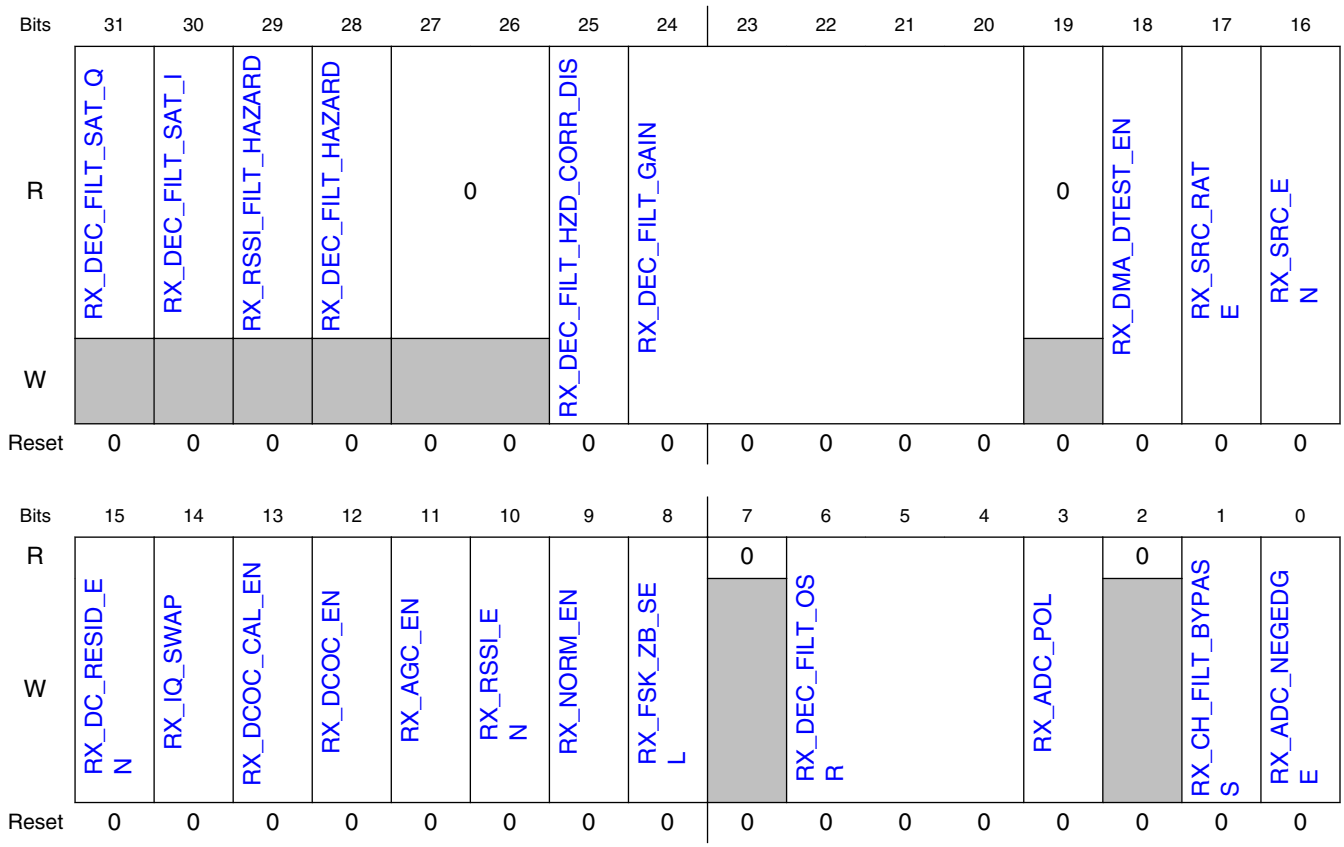
Offset	Register	Width (In bits)	Access	Reset value
1CCh	Receive Channel Filter Coefficient 11 (RX_CHF_COEF_11)	32	RW	0000_0000h
1D0h	AGC Manual AGC Index (AGC_MAN_AGC_IDX)	32	RW	0000_0000h
1D4h	DC Residual Control (DC_RESID_CTRL)	32	RW	0000_0000h
1D8h	DC Residual Estimate (DC_RESID_EST)	32	RO	0000_0000h
1DCh	RX RC Calibration Control0 (RX_RCCAL_CTRL0)	32	RW	0000_0000h
1E0h	RX RC Calibration Control1 (RX_RCCAL_CTRL1)	32	RW	0000_0000h
1E4h	RX RC Calibration Status (RX_RCCAL_STAT)	32	RO	0210_4210h
1E8h	Aux PLL Frequency Calibration Control (AUXPLL_FCAL_CTRL)	32	RW	0040_0000h
1ECh	Aux PLL Frequency Calibration Count 6 (AUXPLL_FCAL_CNT6)	32	RO	0000_0000h
1F0h	Aux PLL Frequency Calibration Count 5 and 4 (AUXPLL_FCAL_CNT5_4)	32	RO	0000_0000h
1F4h	Aux PLL Frequency Calibration Count 3 and 2 (AUXPLL_FCAL_CNT3_2)	32	RO	0000_0000h
1F8h	Aux PLL Frequency Calibration Count 1 and 0 (AUXPLL_FCAL_CNT1_0)	32	RO	0000_0000h

45.3.3.3.1.2 RX Digital Control (RX_DIG_CTRL)

45.3.3.3.1.2.1 Offset

Register	Offset
RX_DIG_CTRL	0h

45.3.3.3.1.2.2 Diagram



45.3.3.3.1.2.3 Fields

Field	Function
31 RX_DEC_FILT_SAT_Q	Decimator output, saturation detected for Q channel This bit will be set if a saturation condition is detected in the decimator filter Q channel. This bit is cleared by tsm_rx_dcoc_init and rx_init. 0b - A saturation condition has not occurred. 1b - A saturation condition has occurred.
30 RX_DEC_FILT_SAT_I	Decimator output, saturation detected for I channel This bit will be set if a saturation condition is detected in the decimator filter I channel. This bit is cleared by tsm_rx_dcoc_init and rx_init. 0b - A saturation condition has not occurred. 1b - A saturation condition has occurred.
29 RX_RSSI_FILT_HAZARD	Decimator output for RSSI, hazard condition detected This bit will be set if a hazard condition is detected in either the I or Q decimator filter related to the wideband RSSI measurement. This does not indicate that a data corruption has occurred, only that the conditions associated with data corruption were detected. This bit is cleared by rx_init. 0b - A hazard condition has not been detected 1b - A hazard condition has been detected
28	Decimator output, hazard condition detected

Table continues on the next page...

Field	Function
RX_DEC_FILT_HAZARD	This bit will be set if a hazard condition is detected in either the I or Q decimator filter. This does not indicate that a data corruption has occurred, only that the conditions associated with data corruption were detected. This bit is cleared by rx_init. 0b - A hazard condition has not been detected 1b - A hazard condition has been detected
27-26 —	Reserved.
25 RX_DEC_FILT_HZD_CORR_DIS	Decimator filter hazard correction disable This bit should be set for normal operation.
24-20 RX_DEC_FILT_GAIN	Decimation Filter Fractional Gain Defines the fractional gain which is applied at the decimator output. The format is u0.5. The gain applied is 1 + RX_DEC_FILT_GAIN/32, so e.g. if RX_DEC_FILT_GAIN is 5'b10110=5'd22, the gain is 1+22/32=1.6875. The nominal gain through the ADC is 2048codes/1.7V; the combined gain through the ADC and decimator would therefore be 2048codes/1.7e3mV*1.6875=2.03codes/mV for the example RX_DEC_FILT_GAIN=5'b10110.
19 —	Reserved.
18 RX_DMA_DTEST_EN	RX DMA and DTEST enable This bit should be set to ensure that all of the rx_dig outputs related to DMA or DTEST are enabled. In mission mode this bit is intended to be cleared to reduce switching power.
17 RX_SRC_RATE	RX Sample Rate Converter Rate Selections 0b - SRC is configured for a First Order Hold rate of 8/13. 1b - SRC is configured for a Zero Order Hold rate of 12/13.
16 RX_SRC_EN	RX Sample Rate Converter Enable 0b - SRC is disabled. 1b - SRC is enabled.
15 RX_DC_RESID_EN	DC Residual Enable Enables DC Residual block 0b - DC Residual block is disabled. 1b - DC Residual block is enabled.
14 RX_IQ_SWAP	RX IQ Swap Enable swap of I/Q channels (does not affect ADC raw mode). 0b - IQ swap is disabled. 1b - IQ swap is enabled.
13 RX_DCOC_CAL_EN	DCOC Calibration Enable Enable DCOC warm-up calibration in receiver. 0b - DCOC calibration is disabled. 1b - DCOC calibration is enabled.
12 RX_DCOC_EN	DCOC Enable Enables DCO calculation and application of corrections. 0b - DCOC is disabled. 1b - DCOC is enabled.
11 RX_AGC_EN	AGC Global Enable Does NOT affect user gains (user gain programming has priority). 0b - AGC is disabled.

Table continues on the next page...

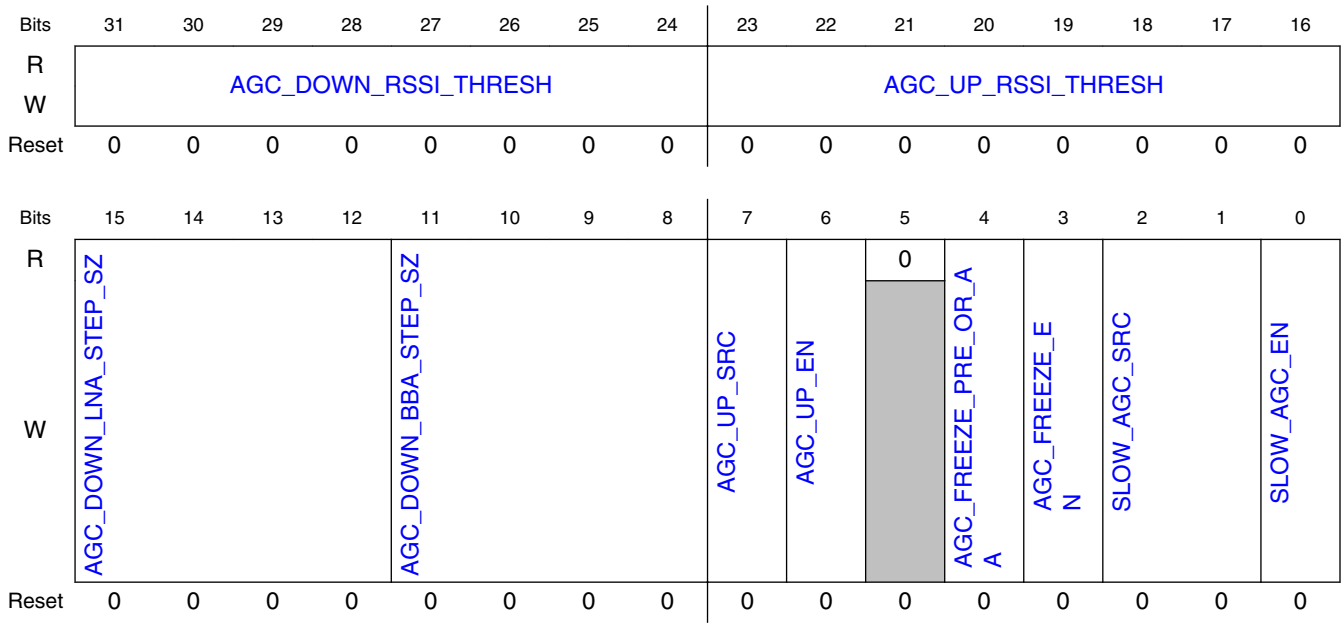
Field	Function
	1b - AGC is enabled.
10 RX_RSSI_EN	RSSI Measurement Enable 0b - RSSI measurement is disabled. 1b - RSSI measurement is enabled.
9 RX_NORM_EN	Normalizer Enable 0b - Normalizer is disabled. 1b - Normalizer is enabled.
8 RX_FSK_ZB_SEL	FSK / 802.15.4 demodulator select Select between FSK and 802.15.4 demodulator. This is used in the RSSI and AGC to select trigger source input signals. 0b - FSK demodulator. 1b - 802.15.4 demodulator.
7 —	Reserved.
6-4 RX_DEC_FILT_OSR	Decimation Filter Oversampling **NOTE:** All undocumented values are Reserved. 000b - OSR 4 001b - OSR 8 010b - OSR 16 011b - OSR 6 100b - OSR 32 101b - OSR 12 110b - OSR 24
3 RX_ADC_POL	Receive ADC Polarity Selects polarity of the ADC data 0b - ADC output of 1'b0 maps to -1, 1'b1 maps to +1 (default) 1b - ADC output of 1'b0 maps to +1, 1'b1 maps to -1
2 —	Reserved.
1 RX_CH_FILT_BYPASS	Receive Channel Filter Bypass Selects whether to disable and bypass channel filter. 0b - Channel filter is enabled. 1b - 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. 0b - Register ADC data on positive edge of clock 1b - Register ADC data on negative edge of clock

45.3.3.3.1.3 AGC Control 0 (AGC_CTRL_0)

45.3.3.3.1.3.1 Offset

Register	Offset
AGC_CTRL_0	4h

45.3.3.3.1.3.2 Diagram



45.3.3.3.1.3.3 Fields

Field	Function
31-24 AGC_DOWN_RSSI_THRESH	AGC DOWN RSSI Threshold ADC RSSI threshold to take downward step (AGC slow). If the ADC RSSI measurment is higher than this threshold, a downward gain step can be taken. The ADC RSSI is measured directly from the ADC input and does not have any adjustment applied for gain settings, etc. In other words, it is a measurement of how saturated the digital bus is, similar to a dBFS measurement.
23-16 AGC_UP_RSSI_THRESH	AGC UP RSSI Threshold ADC RSSI threshold to take upward step (AGC slow). If the ADC RSSI measurment is below this threshold, an upward gain step can be taken. The ADC RSSI is measured directly from the ADC input and does not have any adjustment applied for gain settings, etc. In other words, it is a measurement of how saturated the digital bus is, similar to a dBFS measurement.
15-12 AGC_DOWN_LNA_STEP_SZ	AGC_DOWN_LNA_STEP_SZ Number of table steps for downward step (LNA) in AGC fast.
11-8 AGC_DOWN_BBA_STEP_SZ	AGC_DOWN_BBA_STEP_SZ Number of table steps for downward step (BBA) in AGC fast.
7 AGC_UP_SRC	AGC Up Source Criterion to use for upward AGC steps in SLOW state. For pdet_lo, a timing window is observed for no assertions of the LO peak detectors and then an upward step is made. If RSSI is selected, the current RSSI measurement is compared with the UP threshold to determine if an upward step is due. 0b - PDET LO 1b - RSSI

Table continues on the next page...

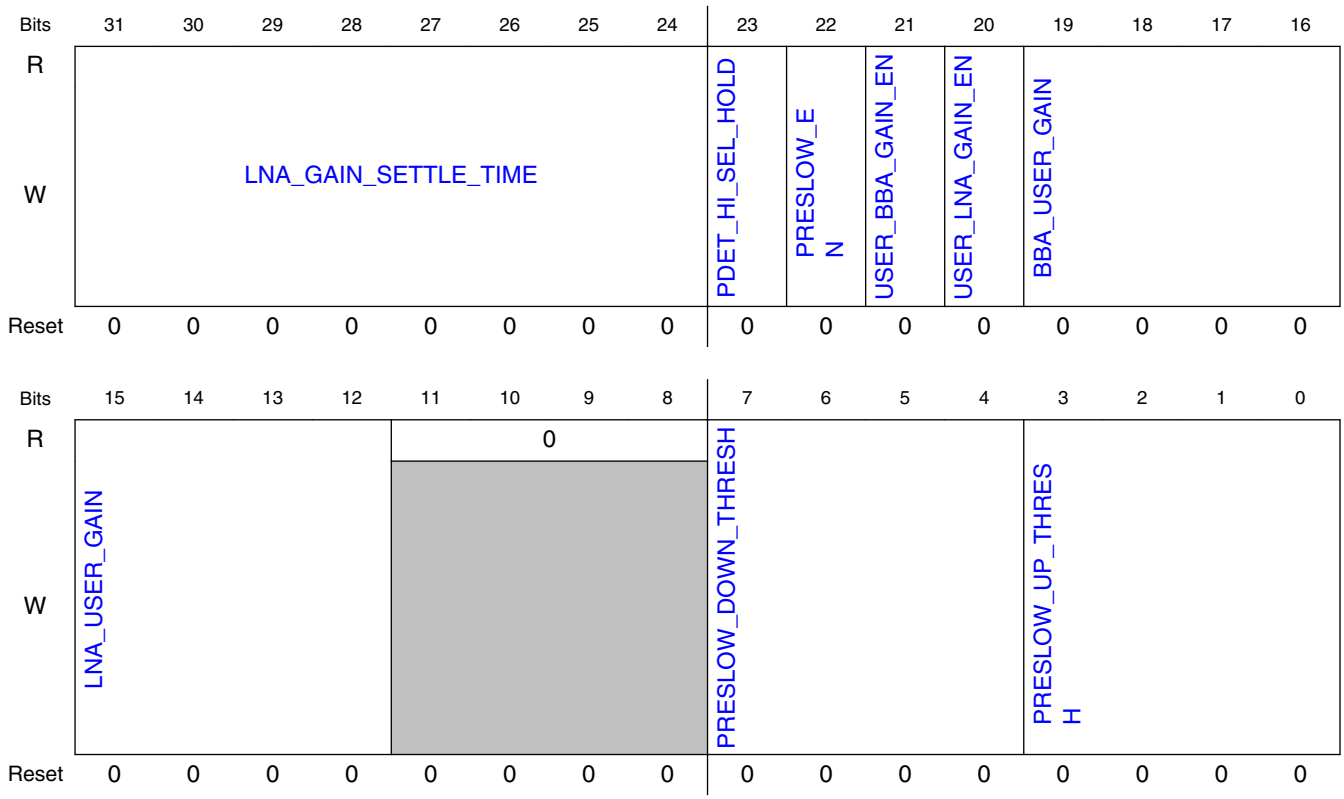
Field	Function
6 AGC_UP_EN	AGC Up Enable Allow AGC to take upward steps in the gain table in slow mode.
5 —	Reserved.
4 AGC_FREEZE_PRE_OR_AA	AGC Freeze Source Selection Select trigger source for entering freeze AGC (HOLD state). 0b - Access Address match (for active protocol) 1b - Preamble Detect (for active protocol)
3 AGC_FREEZE_EN	AGC Freeze Enable Allow AGC to freeze (ie enter HOLD state). 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. For address match and preamble detect, the trigger is generated by the PHY. If the Fast AGC expire timer is selected, when the timer expires, the AGC state machine will transition into SLOW. 00b - Access Address match (for active protocol) 01b - Preamble Detect (for active protocol) 10b - Fast AGC expire timer 11b - Reserved
0 SLOW_AGC_EN	Slow AGC Enable Allow AGC to enter into slow mode.

45.3.3.3.1.4 AGC Control 1 (AGC_CTRL_1)

45.3.3.3.1.4.1 Offset

Register	Offset
AGC_CTRL_1	8h

45.3.3.3.1.4.2 Diagram



45.3.3.3.1.4.3 Fields

Field	Function
31-24 LNA_GAIN_SETTLE_TIME	LNA_GAIN_SETTLE_TIME At LNA gain change, number of clocks to assert TZA peak detector reset (for automatic control). Should be programmed greater than zero.
23 PDET_HI_SEL_HOLD	AGC HOLD hysteresis When enabled, and the AGC state machine is in HOLD, the BBA clip detector threshold is raised by 1 count. 0b - Disabled. 1b - Enabled.
22 PRESLOW_EN	Pre-slow Enable Enable the pre-slow state where signal headroom is checked before entering SLOW. 0b - Pre-slow is disabled. 1b - Pre-slow is enabled.
21 USER_BBA_GAIN_EN	User BBA Gain Enable Enable user defined BBA gain (no AGC).
20 USER_LNA_GAIN_EN	User LNA Gain Enable Enable user defined LNA gain (no AGC).

Table continues on the next page...

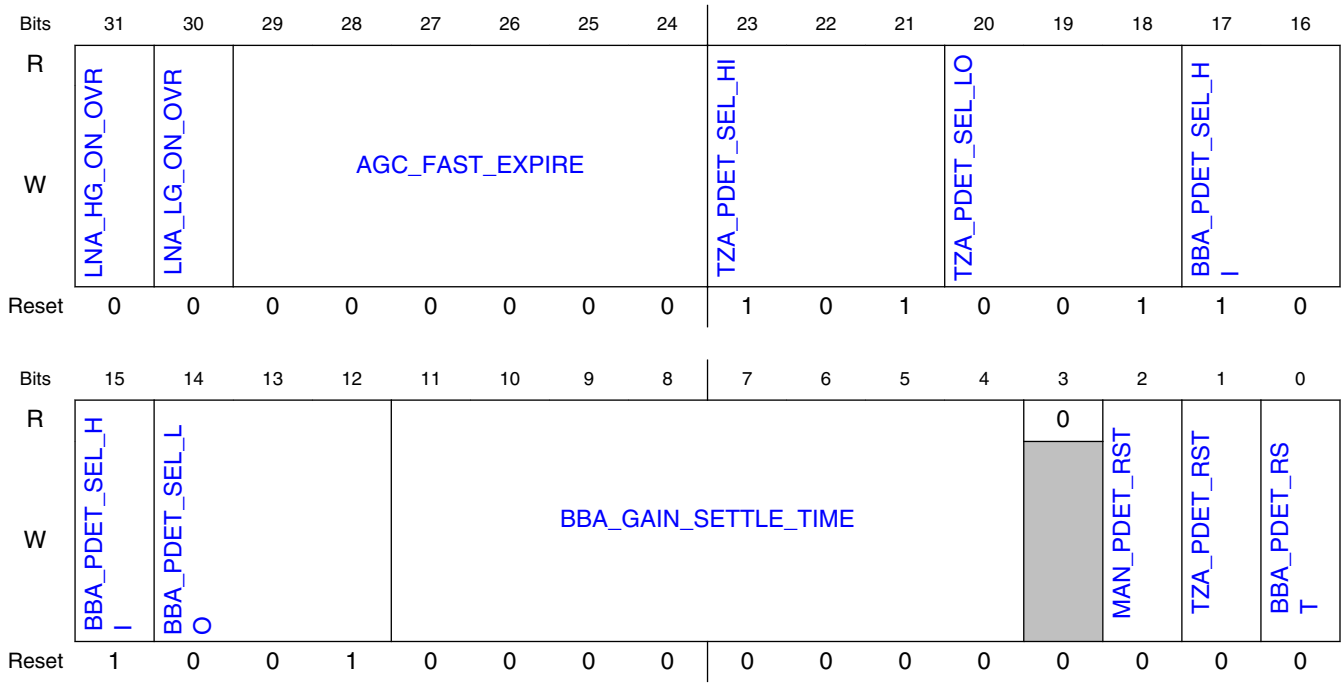
Field	Function
19-16 BBA_USER_GAIN	BBA_USER_GAIN User defined BBA gain index if user_bba_gain_en =1
15-12 LNA_USER_GAIN	LNA_USER_GAIN User defined LNA gain index if user_lna_gain_en =1.
11-8 —	Reserved.
7-4 PRESLOW_DOWN_THRESH	PRESLOW_DOWN_THRESH ADC RSSI threshold to take downward step in (AGC PRESLOW state). The actual threshold is computed as the programmed value minus 10, so the threshold assumes values in the range [-10,5]. If the ADC RSSI measurment is above this threshold, a downward gain step of 1 is taken. The ADC RSSI is measured directly from the ADC input and does not have any adjustment applied for gain settings, etc. In other words, it is a measurement of how saturated the digital bus is, similar to a dBFS measurement.
3-0 PRESLOW_UP_THRESH	PRESLOW_UP_THRESH ADC RSSI threshold to take upward step in (AGC PRESLOW state). The actual threshold is computed as the programmed value minus 15, so the threshold assumes values in the range [-15,0]. If the ADC RSSI measurment is below this threshold, an upward gain step of 1 is taken. The ADC RSSI is measured directly from the ADC input and does not have any adjustment applied for gain settings, etc. In other words, it is a measurement of how saturated the digital bus is, similar to a dBFS measurement.

45.3.3.3.1.5 AGC Control 2 (AGC_CTRL_2)

45.3.3.3.1.5.1 Offset

Register	Offset
AGC_CTRL_2	Ch

45.3.3.3.1.5.2 Diagram



45.3.3.3.1.5.3 Fields

Field	Function
31 LNA_HG_ON_OVR	LNA_HG_ON override If set, the lna high gain "on" signal is always asserted. Otherwise, this signal is asserted automatically based on the LNA_GAIN code.
30 LNA_LG_ON_OVR	LNA_LG_ON override If set, the lna low gain "on" signal is always asserted. Otherwise, this signal is asserted automatically based on the LNA_GAIN code.
29-24 AGC_FAST_EXPIRE	AGC Fast Expire Expire time (uS) for fast AGC (1-63uS).
23-21 TZA_PDET_SEL_HI	TZA PDET Threshold High TZA peak detect HI threshold. 000b - 0.600V 001b - 0.645V 010b - 0.705V 011b - 0.750V 100b - 0.795V 101b - 0.855V 110b - 0.900V 111b - 0.945V
20-18 TZA_PDET_SEL_LO	TZA PDET Threshold Low TZA peak detect LO threshold. 000b - 0.600V

Table continues on the next page...

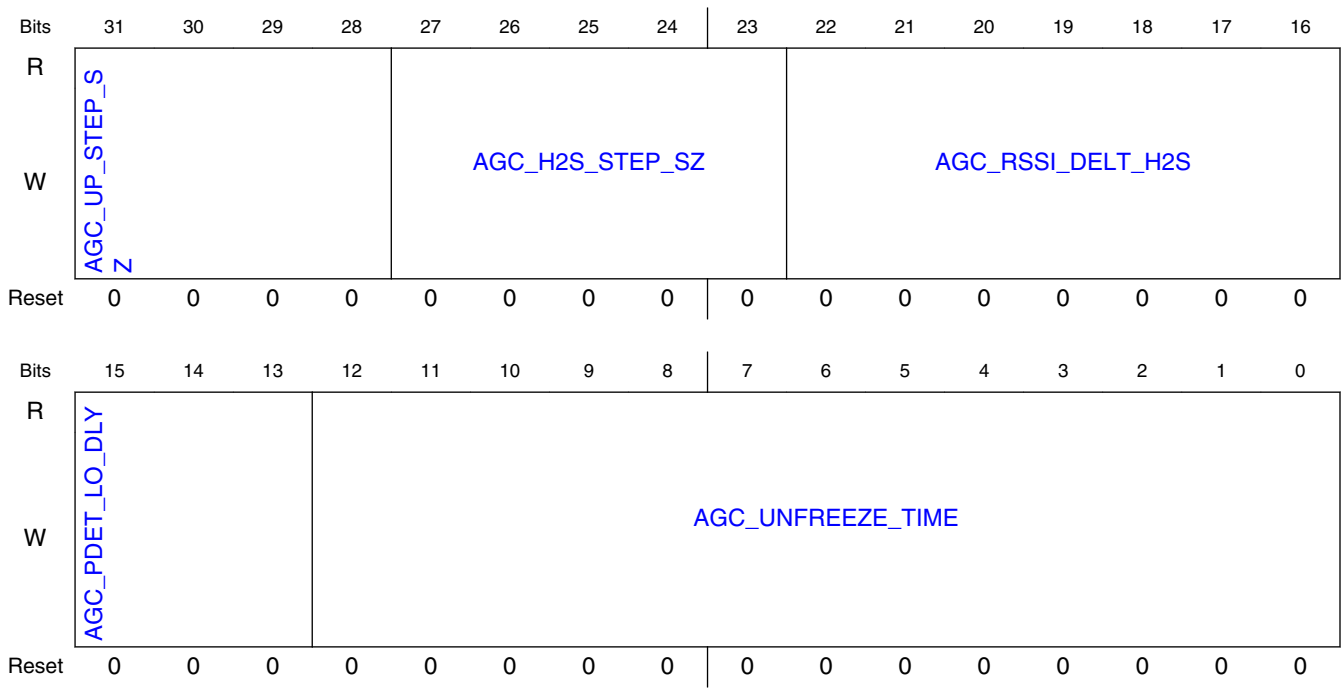
Field	Function
	001b - 0.615V 010b - 0.630V 011b - 0.645V 100b - 0.660V 101b - 0.675V 110b - 0.690V 111b - 0.705V
17-15 BBA_PDET_SEL_HI	BBA PDET Threshold High BBA peak detect HI threshold. 000b - 0.600V 001b - 0.795V 010b - 0.900V 011b - 0.945V 100b - 1.005V 101b - 1.050V 110b - 1.095V 111b - 1.155V
14-12 BBA_PDET_SEL_LO	BBA PDET Threshold Low BBA peak detect LO threshold. 000b - 0.600V 001b - 0.615V 010b - 0.630V 011b - 0.645V 100b - 0.660V 101b - 0.675V 110b - 0.690V 111b - 0.705V
11-4 BBA_GAIN_SETTLE_TIME	BBA Gain Settle Time Number of clocks to assert BBA peak detector reset (for automatic control). Should be programmed greater than zero.
3 —	Reserved.
2 MAN_PDET_RST	MAN PDET Reset 0b - The peak detector reset signals are controlled automatically by the AGC. 1b - The BBA_PDET_RST and TZA_PDET_RST are used to manually control the peak detector reset signals.
1 TZA_PDET_RST	TZA PDET Reset TZA peak detector reset, manual control.
0 BBA_PDET_RST	BBA PDET Reset BBA peak detector reset, manual control.

45.3.3.3.1.6 AGC Control 3 (AGC_CTRL_3)

45.3.3.3.1.6.1 Offset

Register	Offset
AGC_CTRL_3	10h

45.3.3.3.1.6.2 Diagram



45.3.3.3.1.6.3 Fields

Field	Function
31-28 AGC_UP_STEP_SZ	AGC Up Step Size Number of AGC gain table steps for upward step
27-23 AGC_H2S_STEP_SZ	AGC_H2S_STEP_SZ AGC gain table 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. This delta is observed from a previous RSSI measurement to a current measurement.
15-13 AGC_PDET_LO_DLY	AGC Peak Detect Low Delay Time (uS) to wait for pdet low to assert (1-7uS).
12-0	AGC Unfreeze Time

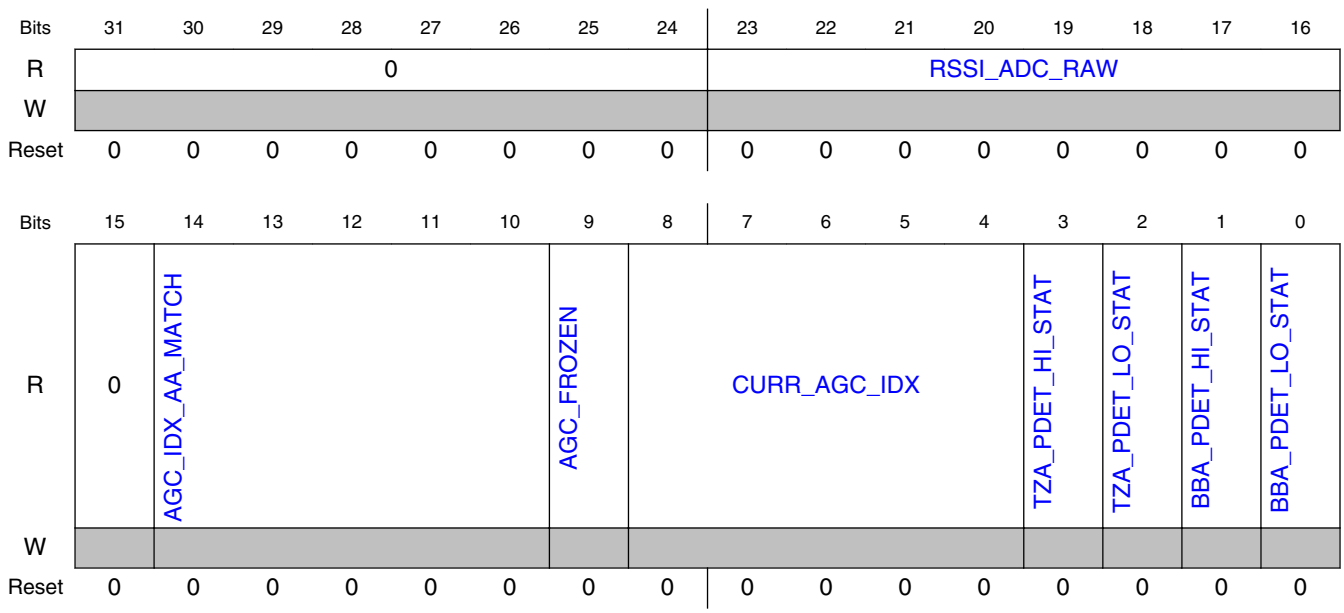
Field	Function
AGC_UNFREEZE_TIME	Time (uS) for AGC to unfreeze (1-8191uS) from HOLD and re-enter SLOW state.

45.3.3.3.1.7 AGC Status (AGC_STAT)

45.3.3.3.1.7.1 Offset

Register	Offset
AGC_STAT	14h

45.3.3.3.1.7.2 Diagram



45.3.3.3.1.7.3 Fields

Field	Function
31-24 —	Reserved.
23-16 RSSI_ADC_RAW	ADC RAW RSSI Reading Reading of ADC rssi (before adjustments)
15	Reserved.

Table continues on the next page...

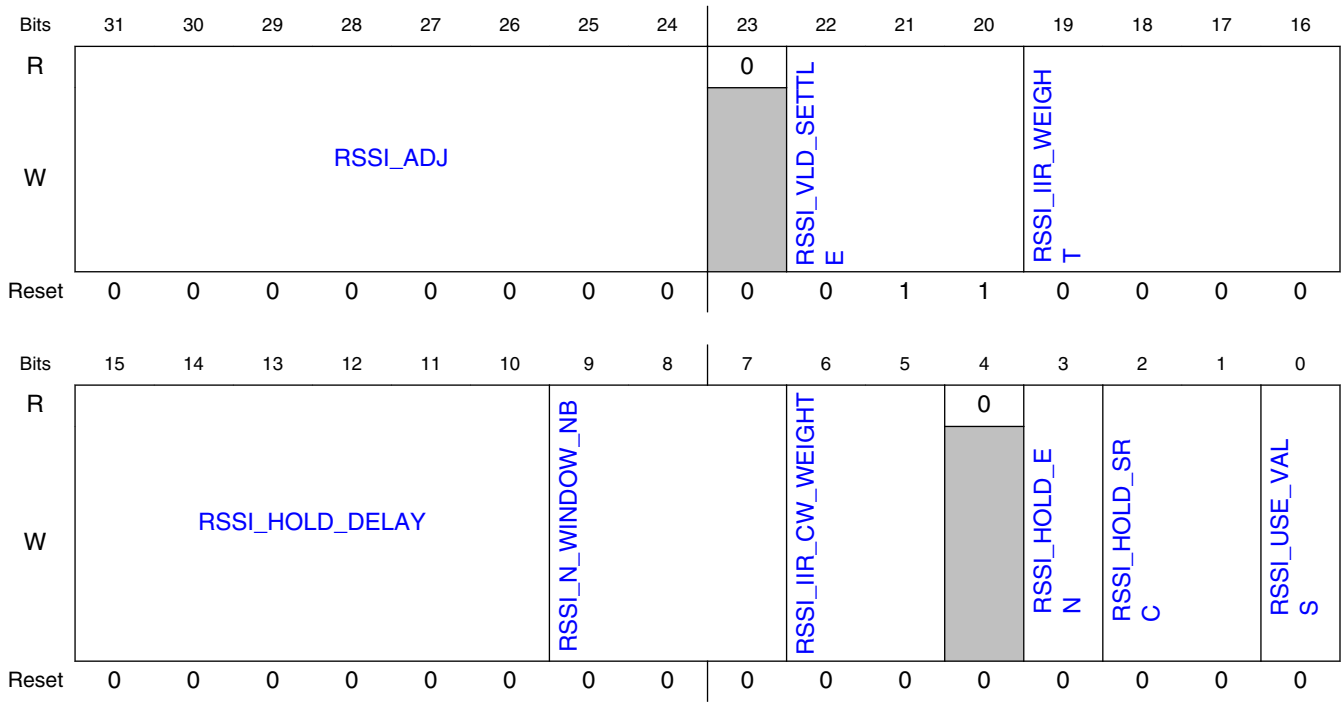
Field	Function
—	
14-10 AGC_IDX_AA_MATCH	AGC Gain Index at AA Match AGC Gain Index captured at AA Match. This value remains static so it can be read by software after the end of the RX burst. The value is cleared to 0 at TSM rx_init and radio reset conditions.
9 AGC_FROZEN	AGC Frozen Status Status of AGC freeze. 0b - AGC is not frozen. 1b - 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 BBA_PDET_HI_STAT	BBA Peak Detector High Status Status of BBA peak detector HI flag (1=set)
0 BBA_PDET_LO_STAT	BBA Peak Detector Low Status Status of BBA peak detector LO flag (1=set)

45.3.3.3.1.8 RSSI Control 0 (RSSI_CTRL_0)

45.3.3.3.1.8.1 Offset

Register	Offset
RSSI_CTRL_0	18h

45.3.3.3.1.8.2 Diagram



45.3.3.3.1.8.3 Fields

Field	Function
31-24 RSSI_ADJ	RSSI Adjustment RSSI calculation adjustment (8-bit signed 1/4 dB).
23 —	Reserved.
22-20 RSSI_VLD_SETTLE	RSSI Valid Settle Sets number of us (times 8) that RSSI will be considered invalid after certain events.
19-16 RSSI_IIR_WEIGHT	RSSI IIR Weighting IIR filter weight for RSSI filtering. **NOTE:** All undocumented values are Reserved. 0000b - Bypass 0001b - 1/2 0010b - 1/4 0011b - 1/8 0100b - 1/16 0101b - 1/32
15-10 RSSI_HOLD_DELAY	RSSI Hold Delay Sets number of us (times 8) that RSSI will run after a hold event before the value is frozen.
9-7	RSSI N Window Average Narrowband

Table continues on the next page...

Field	Function
RSSI_N_WINDOW_NB	Selects Averaging window length for RSSI in Narrowband mode. 000b - No averaging 001b - Averaging window length is 2 samples 010b - Averaging window length is 4 samples 011b - Averaging window length is 8 samples 100b - Averaging window length is 16 samples 101b - Averaging window length is 32 samples
6-5 RSSI_IIR_CW_WEIGHT	RSSI IIR CW Weighting IIR filter weight for RSSI filtering of a CW input. 00b - Bypass 01b - 1/8 10b - 1/16 11b - 1/32
4 —	Reserved.
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	RSSI Hold Source Selection Select trigger source for freezing RSSI measurement. 00b - Access Address match 01b - Preamble Detect 10b - Reserved 11b - 802.15.4 LQI done (1=freeze, 0=run AGC)
0 RSSI_USE_VALS	RSSI Values Selection Enable use of LNA and BBA gain values programmed in registers for calculation.

45.3.3.3.1.9 RSSI Control 1 (RSSI_CTRL_1)

45.3.3.3.1.9.1 Offset

Register	Offset
RSSI_CTRL_1	1Ch

45.3.3.3.1.9.2 Diagram

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

45.3.3.3.1.9.3 Fields

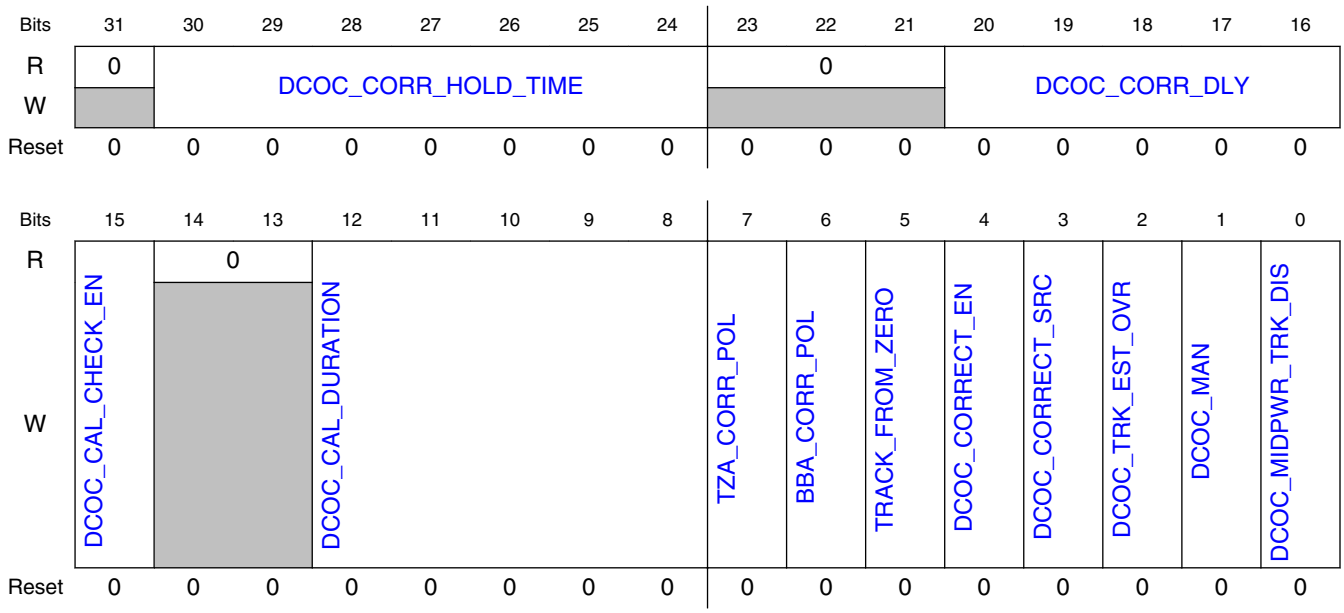
Field	Function
31-24 RSSI_OUT	RSSI Reading RSSI output (8-bit signed).
23-0 —	Reserved.

45.3.3.3.1.10 DCOC Control 0 (DCOC_CTRL_0)

45.3.3.3.1.10.1 Offset

Register	Offset
DCOC_CTRL_0	24h

45.3.3.3.1.10.2 Diagram



45.3.3.3.1.10.3 Fields

Field	Function
31 —	Reserved.
30-24 DCOC_CORR_HOLD_TIME	DCOC Correction Hold Time Delay from last gain change to freezing the DC correction. 0000000b - Reserved 0000001-1111110b - For a 32MHz reference clock, this is the delay in microseconds; for other reference clock frequencies, the delay is scaled accordingly. 1111111b - The DC correction is not frozen.
23-21 —	Reserved.
20-16 DCOC_CORR_DLY	DCOC Correction Delay Wait time between corrections. 00000b - Reserved 00001-11111b - For a 32MHz reference clock, this is the wait time in microseconds; for other reference clock frequencies, the delay is scaled accordingly.
15 DCOC_CAL_CHECK_EN	DCOC Calibration Check Enable 0b - Calibration checking disabled. The DCOC_OFFSET_n registers are always updated during calibration. 1b - Calibration checking enabled. The DCOC_OFFSET_n registers are updated conditionally depending on the outcome of the pass/fail threshold checks performed on the alpha-hat and beta-hat estimates during calibration.
14-13 —	Reserved.
12-8	DCOC Calibration Duration

Table continues on the next page...

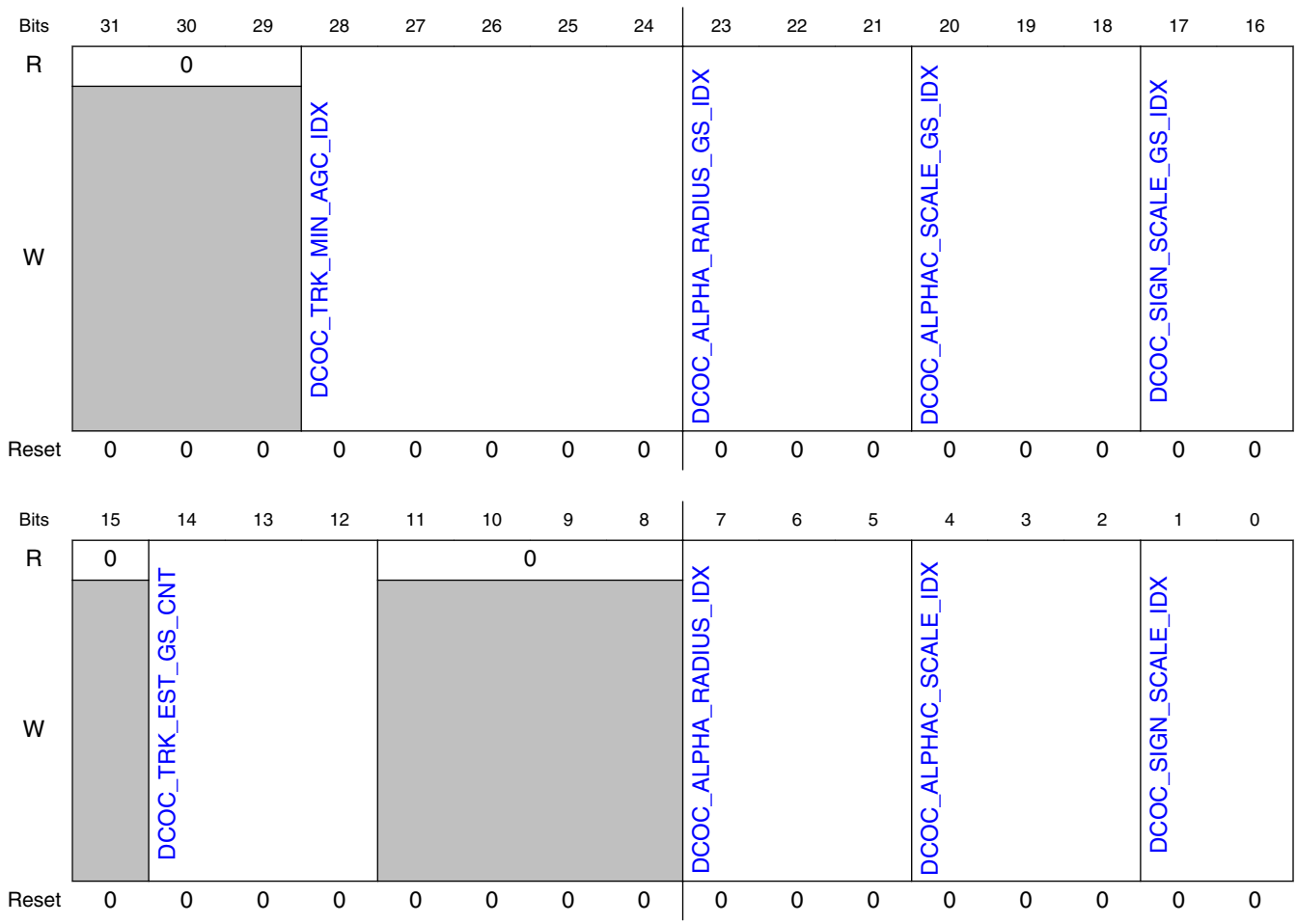
Field	Function
DCOC_CAL_DURATION	00000b - Reserved 00001-11111b - For a 32MHz reference clock, this is the calibration duration in microseconds; for other reference clock frequencies, the delay is scaled accordingly.
7 TZA_CORR_POL	TZA Correction Polarity Selects polarity of TZA corrections. 0b - Normal polarity. 1b - Negative polarity. This should be set if the ADC output is inverted, or if the TZA DACs were implemented with negative polarity.
6 BBA_CORR_POL	BBA Correction Polarity Selects polarity of BBA corrections. 0b - Normal polarity. 1b - Negative polarity. This should be set if the ADC output is inverted, or if the BBA DACs were implemented with negative polarity.
5 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. 0b - Track from current I/Q sample. 1b - Track from zero.
4 DCOC_CORRECT_EN	DCOC Correction Enable 0b - Correction disabled. The DCOC will not correct the DC offset. 1b - Correction enabled. The DCOC will use the TZA and BBA DACs, and apply digital corrections (if DCOC_CORRECT_SRC=1) to correct the DC offset.
3 DCOC_CORRECT_SRC	DCOC Corrector Source If not set, the corrector uses only the DCOC calibration table to apply corrections to the DCOC DACs. 0b - If correction is enabled, the DCOC will use only the DCOC calibration table to correct the DC offset. 1b - If correction is enabled, the DCOC will use the DCOC calibration table and then the tracking estimator to correct the DC offset.
2 DCOC_TRK_EST_OVR	Override for the DCOC tracking estimator 0b - The tracking estimator is enabled only as needed by the corrector 1b - The tracking estimator remains enabled whenever the DCOC is active
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 DCOC_DAC_INIT and DCOC_DIG_MAN, respectively.
0 DCOC_MIDPWR_TRK_DIS	DCOC Mid Power Tracking Disable Disables tracking correction at mid power levels, as indicated by the TZA and BBA lo peak detectors. This is implemented by reseting the counters associated with DCOC_CORR_DLY and DCOC_CORR_HOLD_TIME, so if the lo peak detectors do not continue to assert, tracking corrections would resume operation. The tracking estimator is not disabled and the counter associated with DCOC_TRK_EST_GS_CNT is not reset when this condition occurs. 0b - Tracking corrections are enabled as determined by DCOC_CORRECT_SRC and DCOC_TRK_MIN_AGC_IDX. 1b - Tracking corrections are disabled when either the TZA or BBA lo peak detector asserts.

45.3.3.3.1.11 DCOC Control 1 (DCOC_CTRL_1)

45.3.3.3.1.11.1 Offset

Register	Offset
DCOC_CTRL_1	28h

45.3.3.3.1.11.2 Diagram



45.3.3.3.1.11.3 Fields

Field	Function
31-29 —	Reserved.
28-24 DCOC_TRK_MIN_AGC_IDX	DCOC Tracking Minimum AGC Table Index Specifies the minimum AGC table index value at which tracking is enabled. E.g., if this is 5'd0, then tracking is enabled for all AGC gain table indexes (assuming DCOC_CORRECT_SRC is set) if this is

Table continues on the next page...

Field	Function
	5'd20, then tracking is enabled for AGC gain table indexes 20-26 (assuming DCOC_CORRECT_SRC is set).
23-21 DCOC_ALPHA_RADIUS_GS_IDX	Alpha-R Scaling for Gearshift DCOC Alpha-R Scaling for Gearshift. Radius stepsize used in the DCOC tracking estimator after the gearshift counter has expired. Used when DCOC_CORRECT_SRC=1. 000b - 1 001b - 1/2 010b - 1/4 011b - 1/8 100b - 1/16 101b - 1/32 110b - 1/64 111b - Reserved
20-18 DCOC_ALPHAC_SCALE_GS_IDX	DCOC Alpha-C Scaling for Gearshift DCOC Alpha-C Scaling for Gearshift. I/Q center stepsize used in the DCOC tracking estimator after the gearshift counter has expired. Used when DCOC_CORRECT_SRC=1. 000b - 1/2 001b - 1/4 010b - 1/8 011b - 1/16 100b - 1/32 101b - 1/64 110b - Reserved 111b - Reserved
17-16 DCOC_SIGN_SCALE_GS_IDX	DCOC Sign Scaling for Gearshift DCOC Sign Scaling for Gearshift. Sign()-based scaling factor used in the DCOC tracking estimator after the gearshift counter has expired. Used when DCOC_CORRECT_SRC=1. 00b - 1/8 01b - 1/16 10b - 1/32 11b - 1/64
15 —	Reserved.
14-12 DCOC_TRK_EST_GS_CNT	DCOC Tracking Estimator Gearshift Count Specifies the number of corrections (periods of DCOC_CORR_DLY) before the tracking estimator switches from using the set of parameters {DCOC_ALPHA_RADIUS_IDX, DCOC_ALPHAC_SCALE_IDX, DCOC_SIGN_SCALE_IDX} to the set of gearshift parameters {DCOC_ALPHA_RADIUS_GS_IDX, DCOC_ALPHAC_SCALE_GS_IDX, DCOC_SIGN_SCALE_GS_IDX}. If the value is 0, the set of gearshift parameters are not used.
11-8 —	Reserved.
7-5 DCOC_ALPHA_RADIUS_IDX	Alpha-R Scaling DCOC Alpha-R Scaling. Radius stepsize used in the DCOC tracking estimator. Used when DCOC_CORRECT_SRC=1. 000b - 1 001b - 1/2 010b - 1/4 011b - 1/8 100b - 1/16 101b - 1/32 110b - 1/64

Table continues on the next page...

Field	Function
	111b - Reserved
4-2 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_CORRECT_SRC=1. 000b - 1/2 001b - 1/4 010b - 1/8 011b - 1/16 100b - 1/32 101b - 1/64 110b - Reserved 111b - Reserved
1-0 DCOC_SIGN_SCALE_IDX	DCOC Sign Scaling DCOC Sign Scaling. Sign()-based scaling factor used in the DCOC tracking estimator. Used when DCOC_CORRECT_SRC=1. 00b - 1/8 01b - 1/16 10b - 1/32 11b - 1/64

45.3.3.3.1.12 DCOC DAC Initialization (DCOC_DAC_INIT)

45.3.3.3.1.12.1 Offset

Register	Offset
DCOC_DAC_INIT	2Ch

45.3.3.3.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TZA_DCOC_INIT_Q								TZA_DCOC_INIT_I							
Reset	1	0	0	0	0	0	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	0		BBA_DCOC_INIT_Q						0		BBA_DCOC_INIT_I					
W																
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

45.3.3.3.1.12.3 Fields

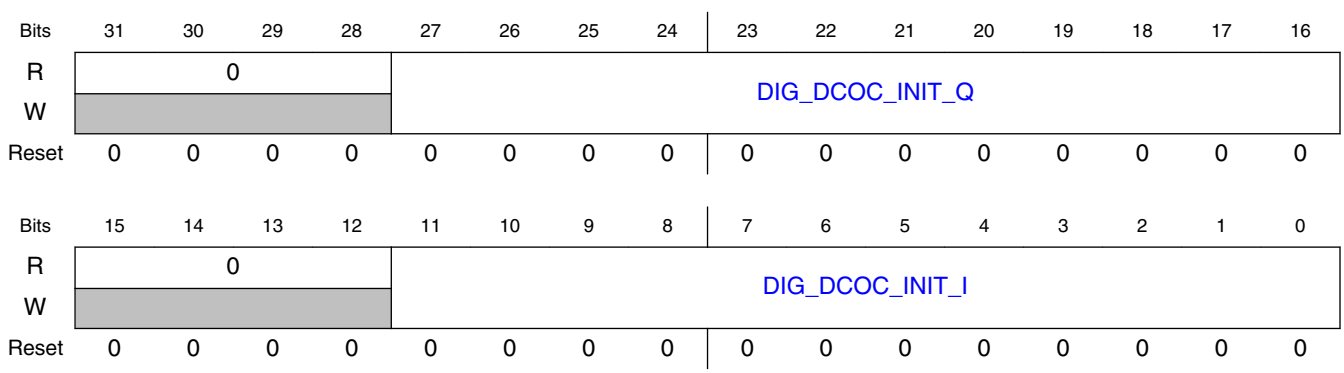
Field	Function
31-24 TZA_DCOC_INIT_Q	DCOC TZA Init Q Value used for DCOC TZA Q channel DAC during calibration and for manual override.
23-16 TZA_DCOC_INIT_I	DCOC TZA Init I Value used for DCOC TZA I channel DAC during calibration and for manual override.
15-14 —	Reserved.
13-8 BBA_DCOC_INIT_Q	DCOC BBA Init Q Value used for DCOC BBA Q channel DAC during calibration and for manual override.
7-6 —	Reserved.
5-0 BBA_DCOC_INIT_I	DCOC BBA Init I Value used for DCOC BBA I channel DAC during calibration and for manual override.

45.3.3.3.1.13 DCOC Digital Correction Manual Override (DCOC_DIG_MAN)

45.3.3.3.1.13.1 Offset

Register	Offset
DCOC_DIG_MAN	30h

45.3.3.3.1.13.2 Diagram



45.3.3.3.1.13.3 Fields

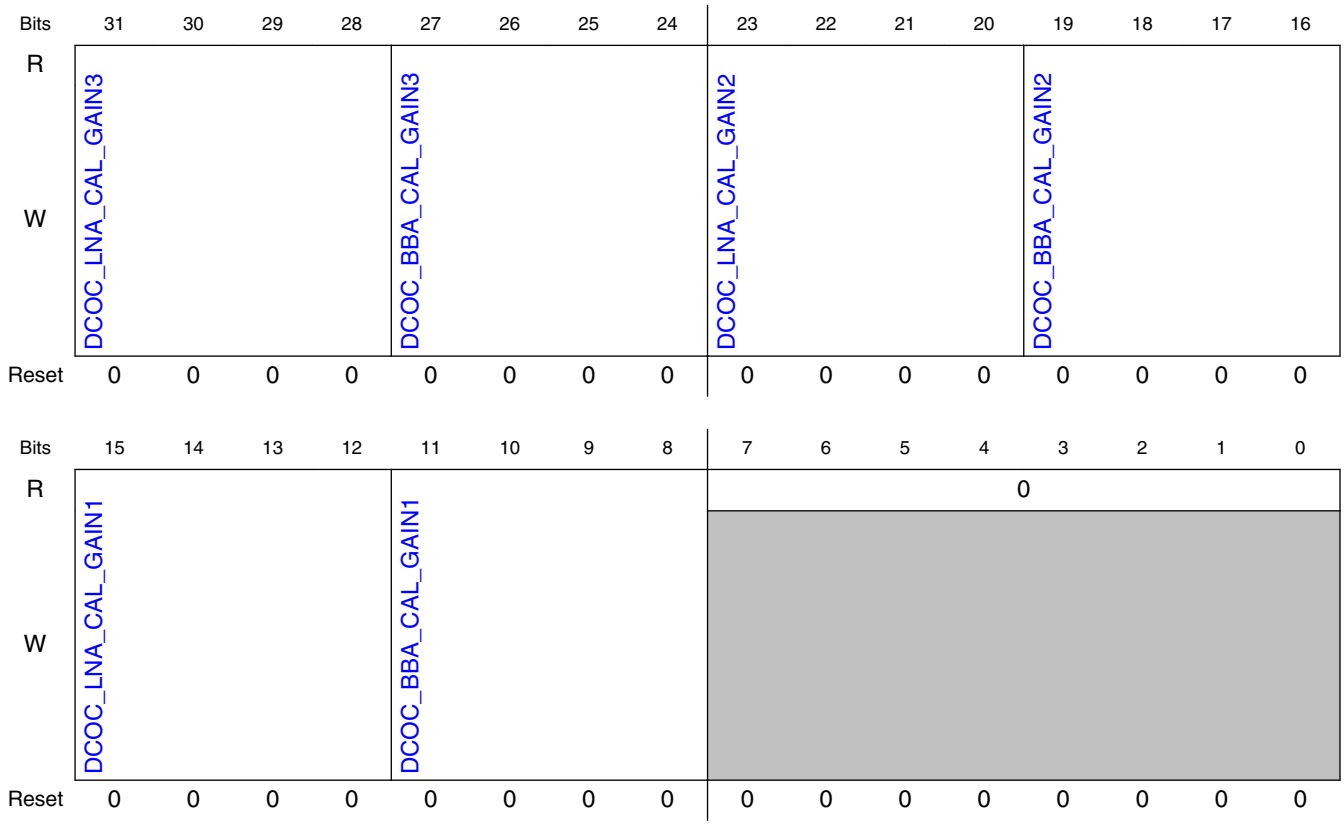
Field	Function
31-28 —	Reserved.
27-16 DIG_DCOC_INIT_Q	DCOC DIG Init Q Manual override for DCOC DIG Q channel correction. Value to be subtracted from downsampled Q channel. Used when DCOC_MAN=1.
15-12 —	Reserved.
11-0 DIG_DCOC_INIT_I	DCOC DIG Init I Manual override for DCOC DIG I channel correction. Value to be subtracted from downsampled I channel. Used when DCOC_MAN=1.

45.3.3.3.1.14 DCOC Calibration Gain (DCOC_CAL_GAIN)

45.3.3.3.1.14.1 Offset

Register	Offset
DCOC_CAL_GAIN	34h

45.3.3.3.1.14.2 Diagram



45.3.3.3.1.14.3 Fields

Field	Function
31-28 DCOC_LNA_CAL_GAIN3	DCOC LNA Calibration Gain 3 The LNA gain index used for the 3rd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
27-24 DCOC_BBA_CAL_GAIN3	DCOC BBA Calibration Gain 3 The BBA gain index used for the 3rd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
23-20 DCOC_LNA_CAL_GAIN2	DCOC LNA Calibration Gain 2 The LNA gain index used for the 2nd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
19-16 DCOC_BBA_CAL_GAIN2	DCOC BBA Calibration Gain 2 The BBA gain index used for the 2nd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
15-12 DCOC_LNA_CAL_GAIN1	DCOC LNA Calibration Gain 1 The LNA gain index used for the 1st DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
11-8	DCOC BBA Calibration Gain 1

Table continues on the next page...

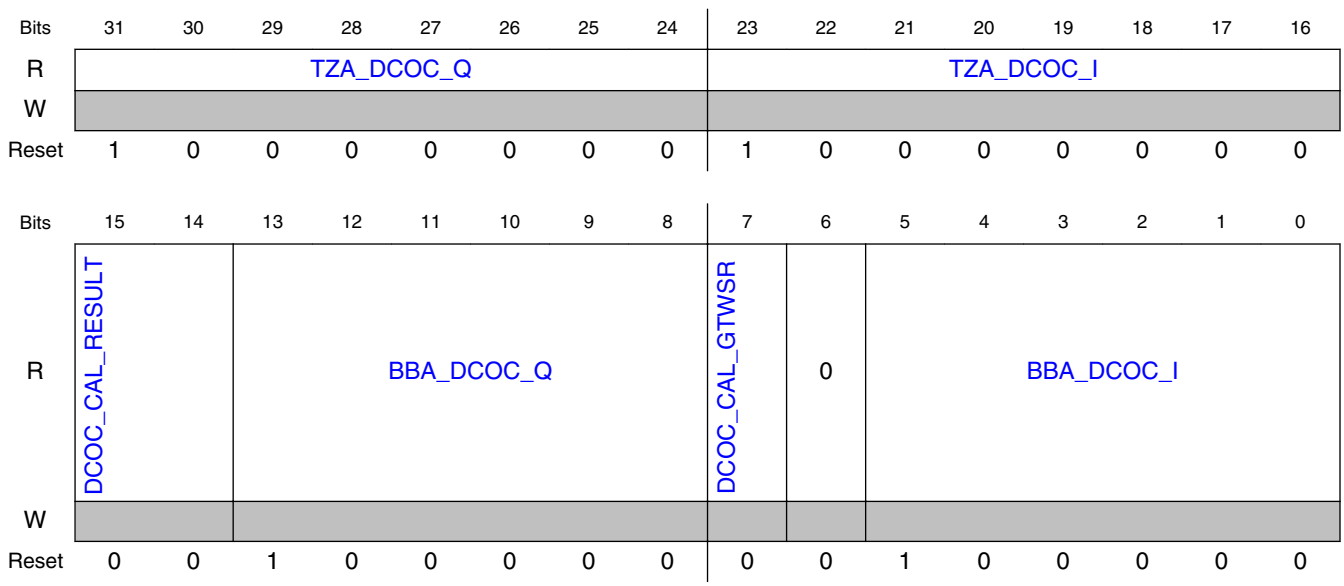
Field	Function
DCOC_BBA_CAL_GAIN1	The BBA gain index used for the 1st DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
7-0 —	Reserved.

45.3.3.3.1.15 DCOC Status (DCOC_STAT)

45.3.3.3.1.15.1 Offset

Register	Offset
DCOC_STAT	38h

45.3.3.3.1.15.2 Diagram



45.3.3.3.1.15.3 Fields

Field	Function
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	DCOC TZA DAC I

Table continues on the next page...

Field	Function
TZA_DCOC_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 DCOC_CAL_RESULT	DCOC_CAL_RESULT Indicates the calibration result. Only applicable when DCOC_CTRL_0[DCOC_CAL_CHECK_EN] is set. 00b - Calibration checks failed. DCOC_OFFSET_n tables not updated. 01b - Calibration checks neither passed nor failed, DCOC_OFFSET_n tables not updated. 10b - Calibration checks neither passed nor failed, DCOC_OFFSET_n tables updated since no previous Pass condition has occurred since the last radio reset. 11b - Calibration checks passed. DCOC_OFFSET_n tables updated
13-8 BBA_DCOC_Q	DCOC BBA DAC Q Current BBA DAC setting for Q channel. Note that the BBA 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 DCOC_CAL_GTWSR	DCOC calibration Good Table Written Since Reset Indicates whether a Passing calibration check has occurred since the last reset. Only applicable when DCOC_CTRL_0[DCOC_CAL_CHECK_EN] is set. 0b - A Passing calibration result has not occurred since the last radio reset. 1b - A Passing calibration result has occurred since the last radio reset.
6 —	Reserved.
5-0 BBA_DCOC_I	DCOC BBA DAC I Current BBA DAC setting for I channel. Note that the BBA 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.

45.3.3.3.1.16 DCOC DC Estimate (DCOC_DC_EST)

45.3.3.3.1.16.1 Offset

Register	Offset
DCOC_DC_EST	3Ch

45.3.3.3.1.16.2 Diagram

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

45.3.3.3.1.16.3 Fields

Field	Function
31-28 —	Reserved.
27-16 DC_EST_Q	DCOC DC Estimate Q Reflects the current DCOC DC tracking estimate for Q channel. Format s11. Used when DCOC_CORRECT_SRC=1. This is provided for debug and characterization purposes only.
15-12 —	Reserved.
11-0 DC_EST_I	DCOC DC Estimate I Reflects the current DCOC DC tracking estimate for I channel. Format s11. Used when DCOC_CORRECT_SRC=1. This is provided for debug and characterization purposes only.

45.3.3.3.1.17 DCOC Calibration Reciprocals (DCOC_CAL_RCP)

45.3.3.3.1.17.1 Offset

Register	Offset
DCOC_CAL_RCP	40h

45.3.3.3.1.17.2 Diagram

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

45.3.3.3.1.17.3 Fields

Field	Function
31-27 —	Reserved.
26-16 ALPHA_CALC_RECIP	Alpha Calculation Reciprocal DCOC Alpha calculation reciprocal (format: u.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 $round([1.0/((G_{L\_HI} - G_{L\_LO})*G_{B\_LO})]*2^{11})$.
15-11 —	Reserved.
10-0 DCOC_TMP_CALC_RECIP	DCOC Calculation Reciprocal DCOC_tmp calculation reciprocal (format: u1.10). 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 $round([1.0/(G_{B\_HI} - G_{B\_LO})]*2^{10})$.

45.3.3.3.1.18 IQMC Control (IQMC_CTRL)

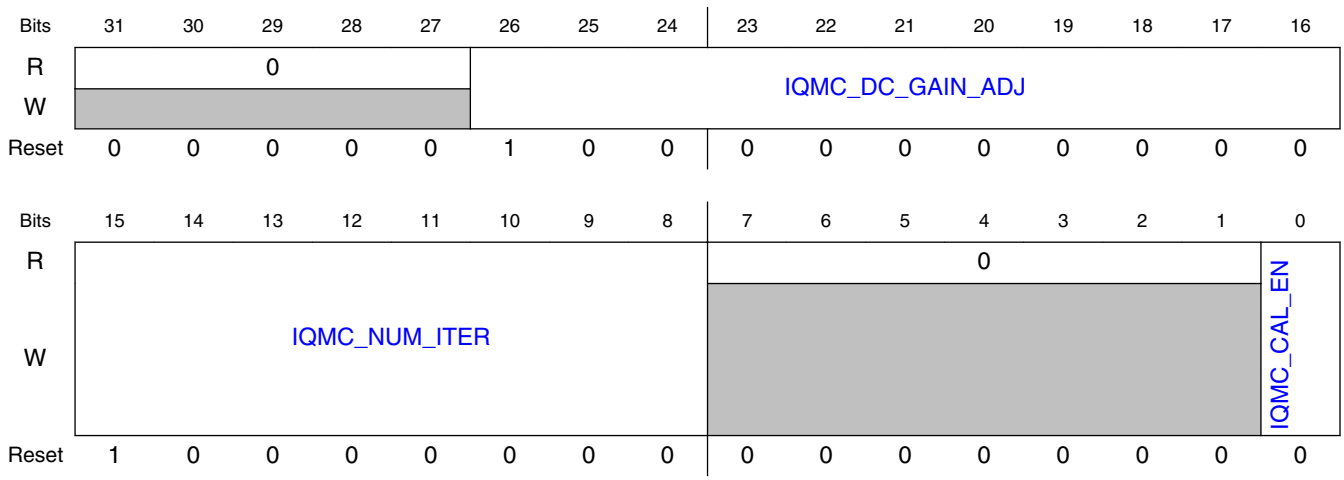
45.3.3.3.1.18.1 Offset

Register	Offset
IQMC_CTRL	48h

45.3.3.3.1.18.2 Function

IQMC Control register. This register can only be accessed when the radio oscillator clock is active.

45.3.3.3.1.18.3 Diagram



45.3.3.3.1.18.4 Fields

Field	Function
31-27 —	Reserved.
26-16 IQMC_DC_GAIN_ADJ	IQ Mismatch Correction DC Gain Coeff I/Q mismatch correction DC gain coefficient. This gain value is used only during DCOC calibration. It is not calculated during the IQMC calibration sequence, so it must be written by software. The format is u1.10; the reset value of 0x400 corresponds to a DC gain coefficient of 1.0.
15-8 IQMC_NUM_ITER	IQ Mismatch Cal Num Iter Number of iterations for IQ Mismatch Calibration.
7-1 —	Reserved.
0 IQMC_CAL_EN	IQ Mismatch Cal Enable Enables IQ mismatch calibration. This bit is self-clearing; it will clear automatically at the end of the calibration sequence.

45.3.3.3.1.19 IQMC Calibration (IQMC_CAL)

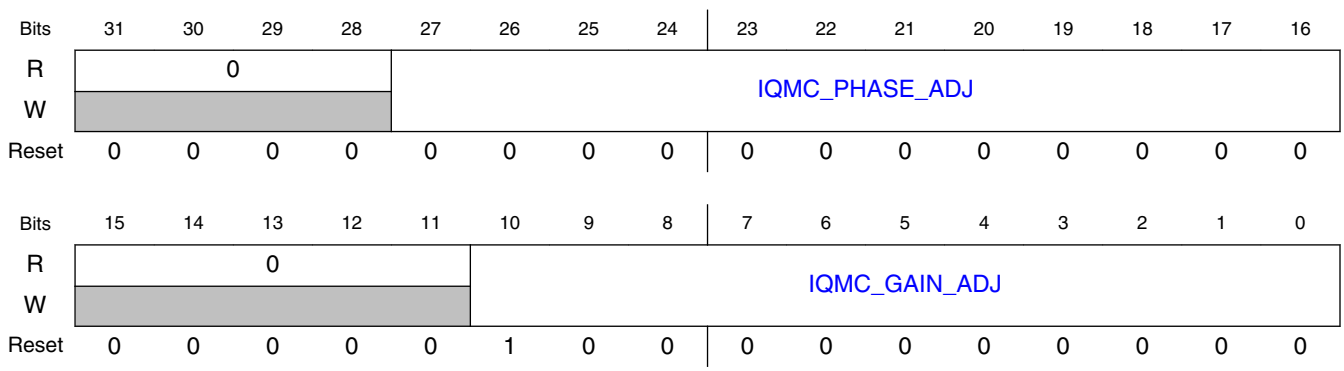
45.3.3.3.1.19.1 Offset

Register	Offset
IQMC_CAL	4Ch

45.3.3.3.1.19.2 Function

IQMC Calibration register. This register can only be accessed when the radio oscillator clock is active.

45.3.3.3.1.19.3 Diagram



45.3.3.3.1.19.4 Fields

Field	Function
31-28 —	Reserved.
27-16 IQMC_PHASE_ADJ	IQ Mismatch Correction Phase Coeff I/Q mismatch correction phase coefficient. This value is updated automatically at the end of the IQMC calibration sequence. It can also be written by software. The format is signed, with the maximum positive value 0x7ff corresponding to a phase coefficient of 0.25, and the maximum negative value 0x800 corresponding to -0.25. The reset value of 0x000 corresponds to a phase coefficient of 0.
15-11 —	Reserved.
10-0 IQMC_GAIN_ADJ	IQ Mismatch Correction Gain Coeff I/Q mismatch correction gain coefficient. This value is updated automatically at the end of the IQMC calibration sequence. It can also be written by software. The format is u1.10. The reset value of 0x400 corresponds to a gain coefficient of 1.0.

45.3.3.3.1.20 LNA_GAIN Step Values 3..0 (LNA_GAIN_VAL_3_0)

45.3.3.3.1.20.1 Offset

Register	Offset
LNA_GAIN_VAL_3_0	50h

45.3.3.3.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_VAL_3								LNA_GAIN_VAL_2							
Reset	0	0	1	1	1	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	LNA_GAIN_VAL_1								LNA_GAIN_VAL_0							
Reset	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0	1

45.3.3.3.1.20.3 Fields

Field	Function
31-24 LNA_GAIN_VAL_3	LNA_GAIN step 3 Gain for LNA gain Step 3, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
23-16 LNA_GAIN_VAL_2	LNA_GAIN step 2 Gain for LNA gain Step 2, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
15-8 LNA_GAIN_VAL_1	LNA_GAIN step 1 Gain for LNA gain Step 1, used in RSSI calculation. Unsigned, 4*(measured_gain_dB+16).
7-0 LNA_GAIN_VAL_0	LNA_GAIN step 0 Gain for LNA gain Step 0, used in RSSI calculation. Unsigned, 4*(measured_gain_dB+16).

45.3.3.3.1.21 LNA_GAIN Step Values 7..4 (LNA_GAIN_VAL_7_4)

45.3.3.3.1.21.1 Offset

Register	Offset
LNA_GAIN_VAL_7_4	54h

45.3.3.3.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_VAL_7								LNA_GAIN_VAL_6							
Reset	1	0	0	0	1	0	1	1	0	1	1	1	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LNA_GAIN_VAL_5								LNA_GAIN_VAL_4							
Reset	0	1	0	1	1	1	0	1	0	1	0	0	1	1	1	1

45.3.3.3.1.21.3 Fields

Field	Function
31-24 LNA_GAIN_VAL_7	LNA_GAIN step 7 Gain for LNA gain Step 7, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
23-16 LNA_GAIN_VAL_6	LNA_GAIN step 6 Gain for LNA gain Step 6, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
15-8 LNA_GAIN_VAL_5	LNA_GAIN step 5 Gain for LNA gain Step 5, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
7-0 LNA_GAIN_VAL_4	LNA_GAIN step 4 Gain for LNA gain Step 4, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).

45.3.3.3.1.22 LNA_GAIN Step Values 8 (LNA_GAIN_VAL_8)

45.3.3.3.1.22.1 Offset

Register	Offset
LNA_GAIN_VAL_8	58h

45.3.3.3.1.22.2 Diagram

Bits	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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LNA_GAIN_VAL_9								LNA_GAIN_VAL_8							
W																
Reset	1	0	1	1	0	1	1	0	1	0	1	0	0	0	0	1

45.3.3.3.1.22.3 Fields

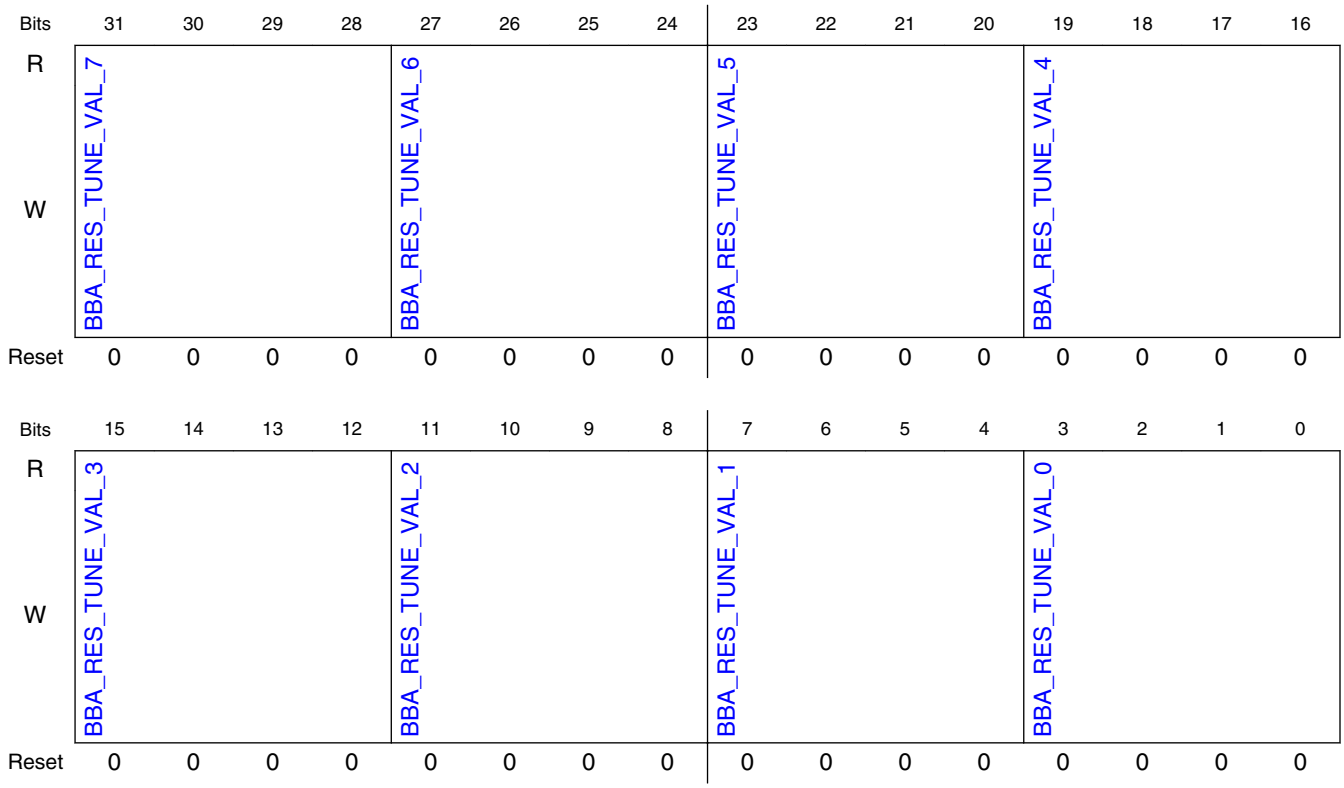
Field	Function
31-16 —	Reserved.
15-8 LNA_GAIN_VAL_9	LNA_GAIN step 9 Gain for LNA gain Step 9, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
7-0 LNA_GAIN_VAL_8	LNA_GAIN step 8 Gain for LNA gain Step 8, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).

45.3.3.3.1.23 BBA Resistor Tune Values 7..0 (BBA_RES_TUNE_VAL_7_0)

45.3.3.3.1.23.1 Offset

Register	Offset
BBA_RES_TUNE_VAL_7_0	5Ch

45.3.3.3.1.23.2 Diagram



45.3.3.3.1.23.3 Fields

Field	Function
31-28 BBA_RES_TUNE_VAL_7	BBA Resistor Tune Step 7 Gain offset for BBA gain Step 7, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
27-24 BBA_RES_TUNE_VAL_6	BBA Resistor Tune Step 6 Gain offset for BBA gain Step 6, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
23-20 BBA_RES_TUNE_VAL_5	BBA Resistor Tune Step 5 Gain offset for BBA gain Step 5, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
19-16 BBA_RES_TUNE_VAL_4	BBA Resistor Tune Step 4 Gain offset for BBA gain Step 4, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
15-12 BBA_RES_TUNE_VAL_3	BBA Resistor Tune Step 3 Gain offset for BBA gain Step 3, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
11-8	BBA Resistor Tune Step 2

Table continues on the next page...

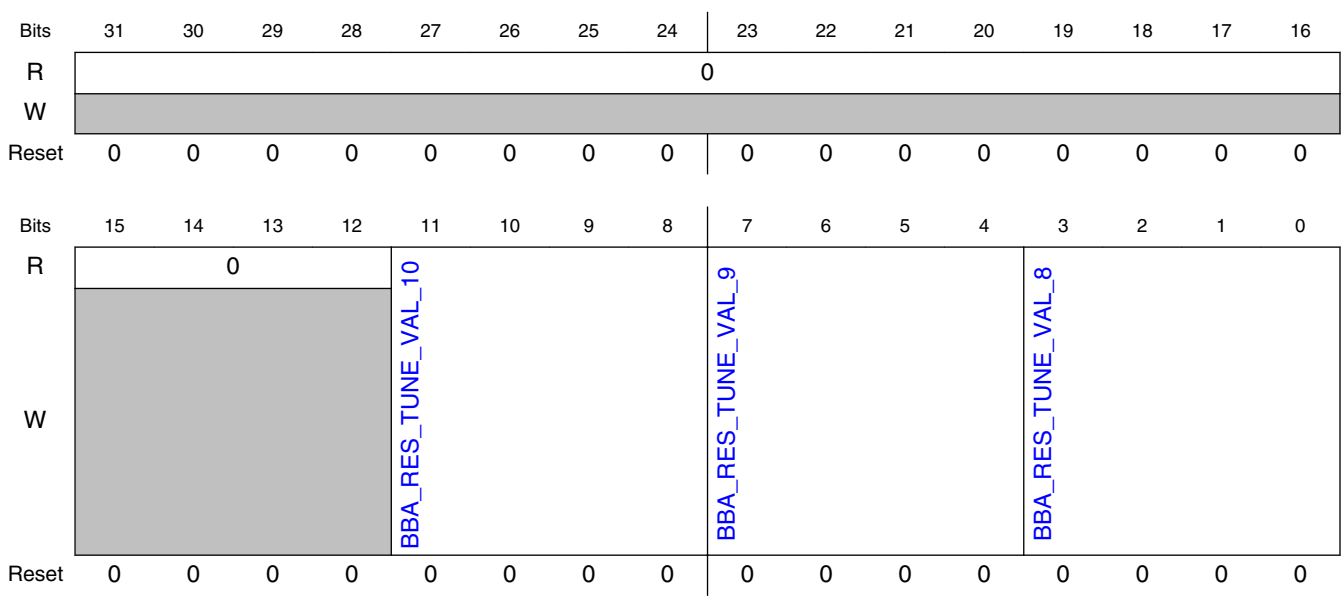
Field	Function
BBA_RES_TUNE_VAL_2	Gain offset for BBA gain Step 2, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
7-4 BBA_RES_TUNE_VAL_1	BBA Resistor Tune Step 1 Gain offset for BBA gain Step 1, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
3-0 BBA_RES_TUNE_VAL_0	BBA Resistor Tune Step 0 Gain offset for BBA gain Step 0, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).

45.3.3.3.1.24 BBA Resistor Tune Values 10..8 (BBA_RES_TUNE_VAL_10_8)

45.3.3.3.1.24.1 Offset

Register	Offset
BBA_RES_TUNE_VAL_10_8	60h

45.3.3.3.1.24.2 Diagram



45.3.3.3.1.24.3 Fields

Field	Function
31-12	Reserved.

Table continues on the next page...

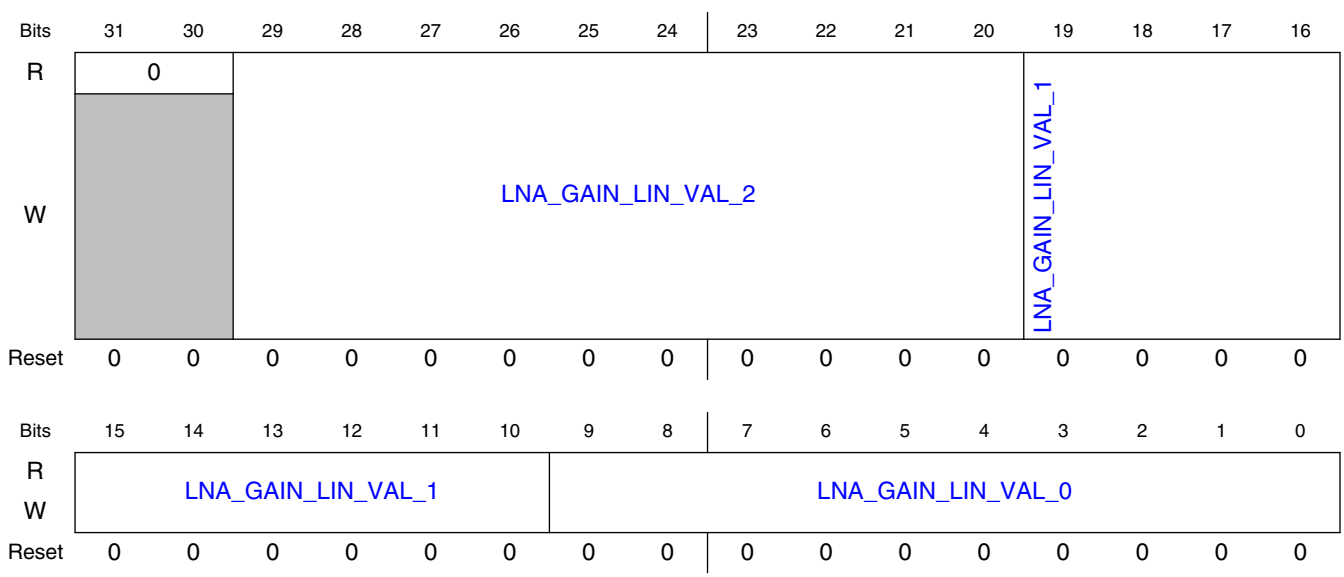
Field	Function
—	
11-8 BBA_RES_TUNE_VAL_10	BBA Resistor Tune Step 10 Gain offset for BBA gain Step 10, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
7-4 BBA_RES_TUNE_VAL_9	BBA Resistor Tune Step 9 Gain offset for BBA gain Step 9, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
3-0 BBA_RES_TUNE_VAL_8	BBA Resistor Tune Step 8 Gain offset for BBA gain Step 8, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).

45.3.3.3.1.25 LNA Linear Gain Values 2..0 (LNA_GAIN_LIN_VAL_2_0)

45.3.3.3.1.25.1 Offset

Register	Offset
LNA_GAIN_LIN_VAL_2_0	64h

45.3.3.3.1.25.2 Diagram



45.3.3.3.1.25.3 Fields

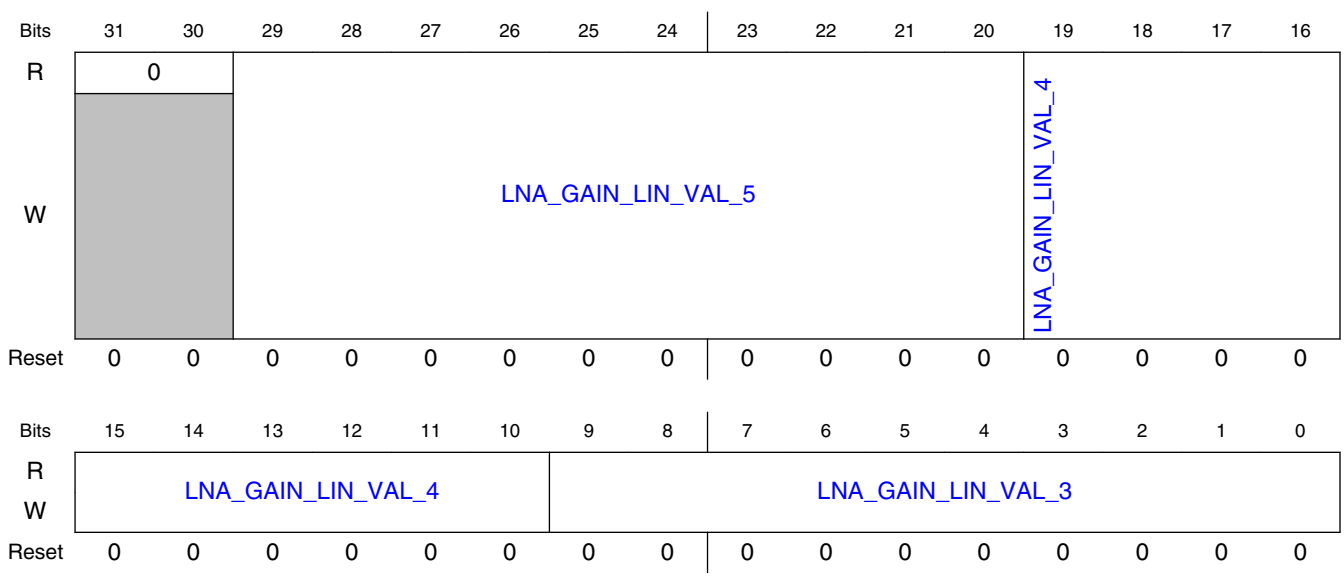
Field	Function
31-30 —	Reserved.
29-20 LNA_GAIN_LIN_VAL_2	LNA Linear Gain Step 2 LNA linear gain value for index 2, e.g. nominal value is 10^(2.2/20). Stored with 2 fractional bits, e.g. round([10^(2.2/20)]*2^2) = 5decimal. Format (8.2).
19-10 LNA_GAIN_LIN_VAL_1	LNA Linear Gain Step 1 LNA linear gain value for index 1, e.g. nominal value is 10^(-3.5/20). Stored with 2 fractional bits, e.g. round([10^(-3.5/20)]*2^2) = 3decimal. Format (8.2).
9-0 LNA_GAIN_LIN_VAL_0	LNA Linear Gain Step 0 LNA linear gain value for index 0, e.g. nominal value is 10^(-8.6/20). Stored with 2 fractional bits, e.g. round([10^(-8.6/20)]*2^2) = 1decimal. Format (8.2).

45.3.3.3.1.26 LNA Linear Gain Values 5..3 (LNA_GAIN_LIN_VAL_5_3)

45.3.3.3.1.26.1 Offset

Register	Offset
LNA_GAIN_LIN_VAL_5_3	68h

45.3.3.3.1.26.2 Diagram



45.3.3.3.1.26.3 Fields

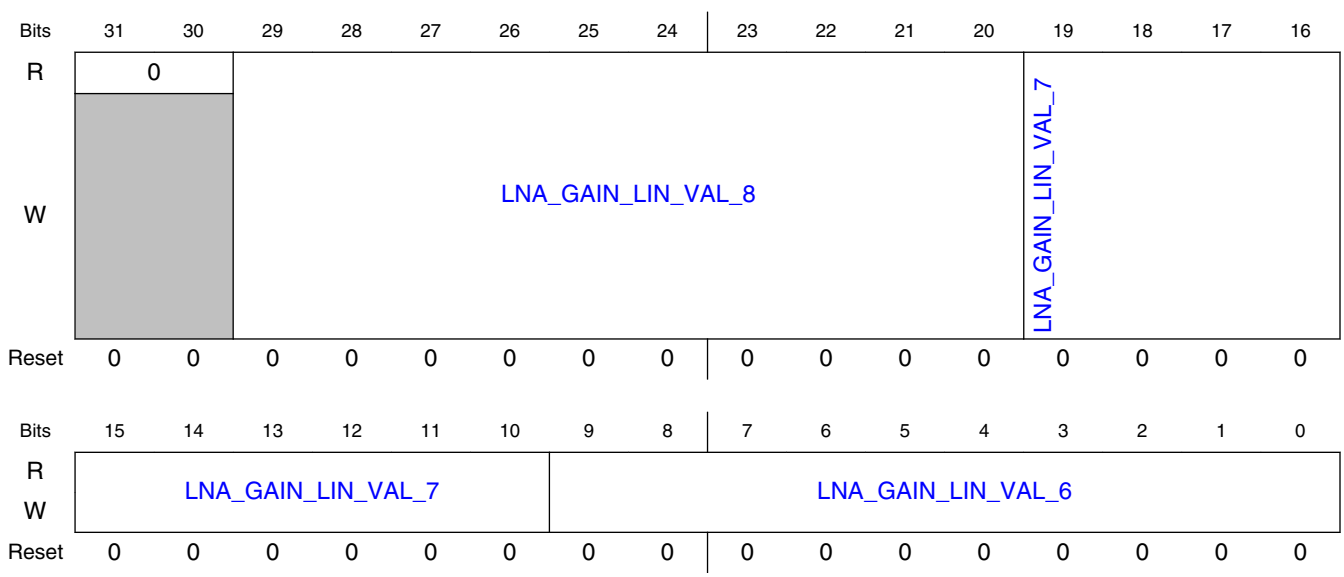
Field	Function
31-30 —	Reserved.
29-20 LNA_GAIN_LIN_VAL_5	LNA Linear Gain Step 5 LNA linear gain value for index 5, e.g. nominal value is 10^(22.7/20). Stored with 2 fractional bits, e.g. round([10^(22.7/20)]*2^2) = 55decimal. Format (8.2).
19-10 LNA_GAIN_LIN_VAL_4	LNA Linear Gain Step 4 LNA linear gain value for index 4, e.g. nominal value is 10^(19.8/20). Stored with 2 fractional bits, e.g. round([10^(19.8/20)]*2^2) = 39decimal. Format (8.2).
9-0 LNA_GAIN_LIN_VAL_3	LNA Linear Gain Step 3 LNA linear gain value for index 3, e.g. nominal value is 10^(13.9/20). Stored with 2 fractional bits, e.g. round([10^(13.9/20)]*2^2) = 20decimal. Format (8.2).

45.3.3.3.1.27 LNA Linear Gain Values 8..6 (LNA_GAIN_LIN_VAL_8_6)

45.3.3.3.1.27.1 Offset

Register	Offset
LNA_GAIN_LIN_VAL_8_6	6Ch

45.3.3.3.1.27.2 Diagram



45.3.3.3.1.27.3 Fields

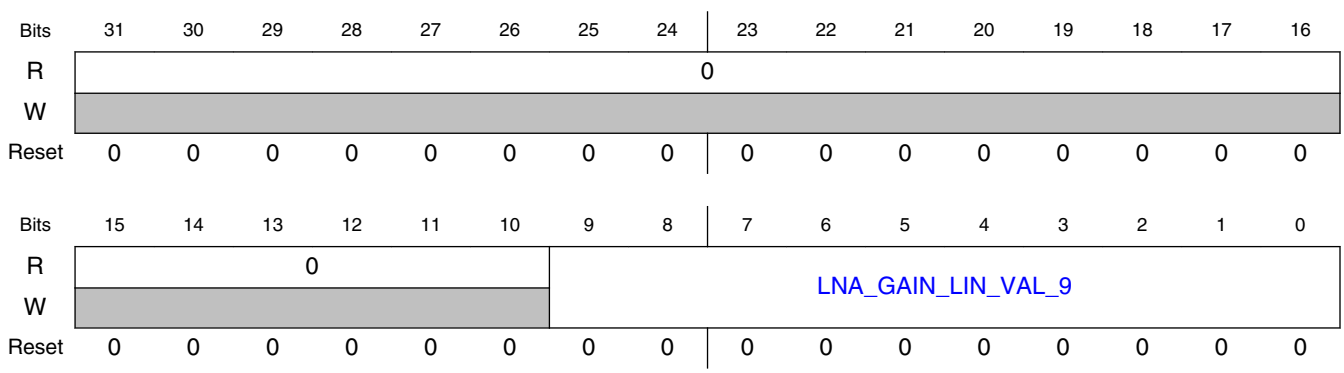
Field	Function
31-30 —	Reserved.
29-20 LNA_GAIN_LIN_VAL_8	LNA Linear Gain Step 8 LNA linear gain value for index 8, e.g. nominal value is 10^(39.9/20). Stored with 2 fractional bits, e.g. round([10^(39.9/20)]*2^2) = 396decimal. Format (8.2).
19-10 LNA_GAIN_LIN_VAL_7	LNA Linear Gain Step 7 LNA linear gain value for index 7, e.g. nominal value is 10^(34.4/20). Stored with 2 fractional bits, e.g. round([10^(34.4/20)]*2^2) = 210decimal. Format (8.2).
9-0 LNA_GAIN_LIN_VAL_6	LNA Linear Gain Step 6 LNA linear gain value for index 6, e.g. nominal value is 10^(28.6/20). Stored with 2 fractional bits, e.g. round([10^(28.6/20)]*2^2) = 108decimal. Format (8.2).

45.3.3.3.1.28 LNA Linear Gain Values 9 (LNA_GAIN_LIN_VAL_9)

45.3.3.3.1.28.1 Offset

Register	Offset
LNA_GAIN_LIN_VAL_9	70h

45.3.3.3.1.28.2 Diagram



45.3.3.3.1.28.3 Fields

Field	Function
31-10 —	Reserved.

Table continues on the next page...

Field	Function
9-0 LNA_GAIN_LIN_VAL_9	LNA Linear Gain Step 9 LNA linear gain value for index 9, e.g. nominal value is 10^(45.4/20). Stored with 2 fractional bits, e.g. round([10^(45.4/20)]*2^2) = 744decimal. Format (8.2).

45.3.3.3.1.29 BBA Resistor Tune Values 3..0 (BBA_RES_TUNE_LIN_VAL_3_0)

45.3.3.3.1.29.1 Offset

Register	Offset
BBA_RES_TUNE_LIN_VAL_3_0	74h

45.3.3.3.1.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BBA_RES_TUNE_LIN_VAL_3								BBA_RES_TUNE_LIN_VAL_2							
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	BBA_RES_TUNE_LIN_VAL_1								BBA_RES_TUNE_LIN_VAL_0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.29.3 Fields

Field	Function
31-24 BBA_RES_TUNE_LIN_VAL_3	BBA Resistor Tune Linear Gain Step 3 BBA linear gain value for index 3 (format: u5.3). Nominal value is 10^(9/20). Stored with 3 fractional bits, e.g. round([10^(9/20)]*2^3) = 23decimal
23-16 BBA_RES_TUNE_LIN_VAL_2	BBA Resistor Tune Linear Gain Step 2 BBA linear gain value for index 2 (format: u5.3). Nominal value is 10^(6/20). Stored with 3 fractional bits, e.g. round([10^(6/20)]*2^3) = 16decimal
15-8 BBA_RES_TUNE_LIN_VAL_1	BBA Resistor Tune Linear Gain Step 1 BBA linear gain value for index 1 (format: u5.3). Nominal value is 10^(3/20). Stored with 3 fractional bits, e.g. round([10^(3/20)]*2^3) = 11decimal
7-0 BBA_RES_TUNE_LIN_VAL_0	BBA Resistor Tune Linear Gain Step 0 BBA linear gain value for index 0 (format: u5.3). Nominal value is 10^(0/20). Stored with 3 fractional bits, e.g. round([10^(0/20)]*2^3) = 8decimal

45.3.3.3.1.30 BBA Resistor Tune Values 7..4 (BBA_RES_TUNE_LIN_VAL_7_4)

45.3.3.3.1.30.1 Offset

Register	Offset
BBA_RES_TUNE_LIN_VAL_7_4	78h

45.3.3.3.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BBA_RES_TUNE_LIN_VAL_7								BBA_RES_TUNE_LIN_VAL_6							
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	BBA_RES_TUNE_LIN_VAL_5								BBA_RES_TUNE_LIN_VAL_4							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.30.3 Fields

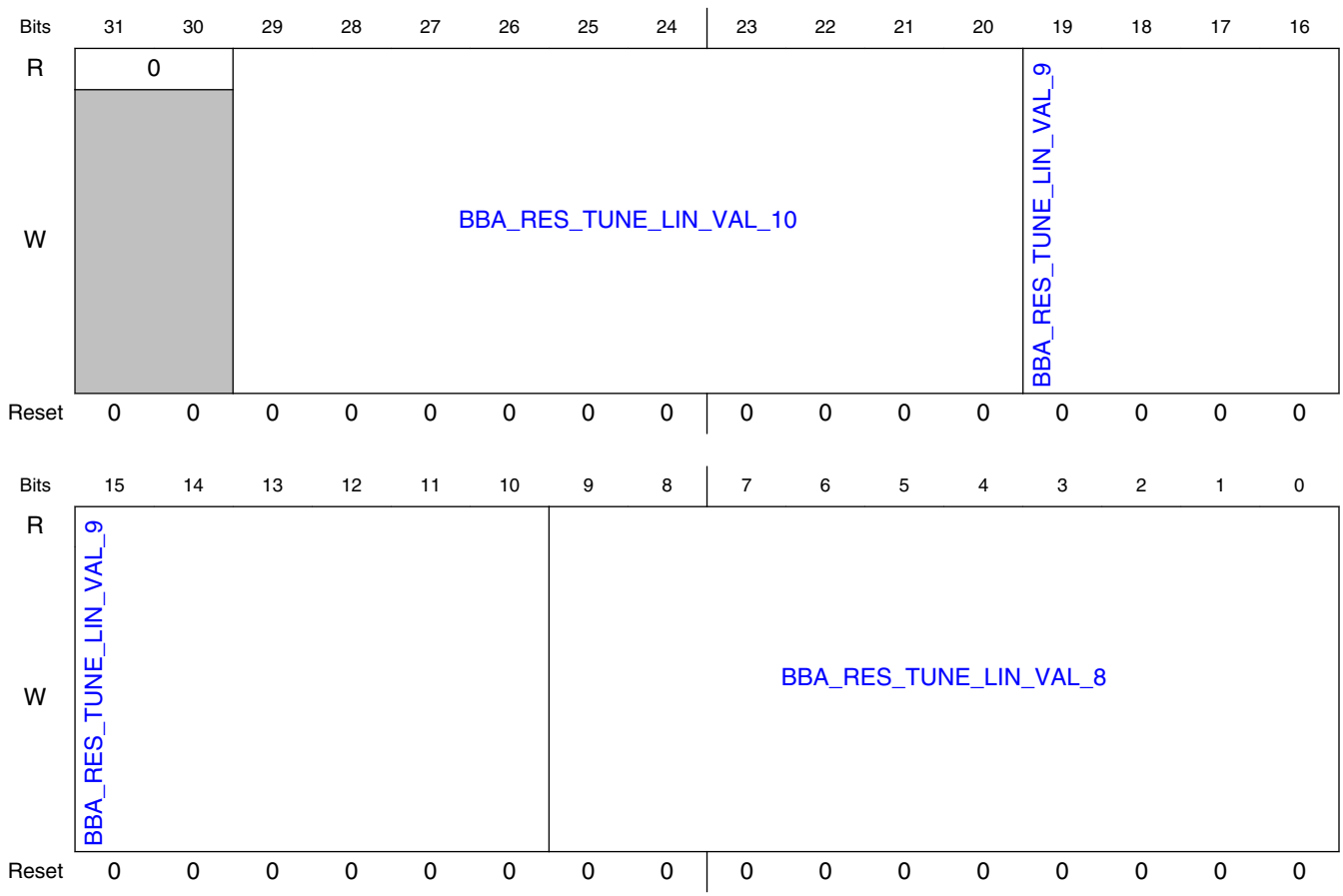
Field	Function
31-24 BBA_RES_TUNE_LIN_VAL_7	BBA Resistor Tune Linear Gain Step 7 BBA linear gain value for index 7 (format: u6.2). Nominal value is 10^(21/20). Stored with 2 fractional bits, e.g. round([10^(21/20)]*2^2) = 45 decimal
23-16 BBA_RES_TUNE_LIN_VAL_6	BBA Resistor Tune Linear Gain Step 6 BBA linear gain value for index 6 (format: u5.3). Nominal value is 10^(18/20). Stored with 3 fractional bits, e.g. round([10^(18/20)]*2^3) = 64decimal
15-8 BBA_RES_TUNE_LIN_VAL_5	BBA Resistor Tune Linear Gain Step 5 BBA linear gain value for index 5 (format: u5.3). Nominal value is 10^(15/20). Stored with 3 fractional bits, e.g. round([10^(15/20)]*2^3) = 45decimal
7-0 BBA_RES_TUNE_LIN_VAL_4	BBA Resistor Tune Linear Gain Step 4 BBA linear gain value for index 4 (format: u5.3). Nominal value is 10^(12/20). Stored with 3 fractional bits, e.g. round([10^(12/20)]*2^3) = 32decimal

45.3.3.3.1.31 BBA Resistor Tune Values 10..8 (BBA_RES_TUNE_LIN_VAL_10_8)

45.3.3.3.1.31.1 Offset

Register	Offset
BBA_RES_TUNE_LIN_VAL_10_8	7Ch

45.3.3.3.1.31.2 Diagram



45.3.3.3.1.31.3 Fields

Field	Function
31-30 —	Reserved.
29-20 BBA_RES_TUNE_LIN_VAL_10	BBA Resistor Tune Linear Gain Step 10 BBA linear gain value for index 10 (format: u7.3). Nominal value is 10^(30/20). Stored with 3 fractional bits, e.g. round([10^(30/20)]*2^3) = 253decimal
19-10	BBA Resistor Tune Linear Gain Step 9

Table continues on the next page...

Field	Function
BBA_RES_TUNE_LIN_VAL_9	BBA linear gain value for index 9 (format: u7.3). Nominal value is 10^(27/20). Stored with 3 fractional bits, e.g. round([10^(27/20)]*2^3) = 179decimal
9-0 BBA_RES_TUNE_LIN_VAL_8	BBA Resistor Tune Linear Gain Step 8 BBA linear gain value for index 8 (format: u7.3). Nominal value is 10^(24/20). Stored with 3 fractional bits, e.g. round([10^(24/20)]*2^3) = 127decimal

45.3.3.3.1.32 AGC Gain Tables Step 03..00 (AGC_GAIN_TBL_03_00)

45.3.3.3.1.32.1 Offset

Register	Offset
AGC_GAIN_TBL_03_00	80h

45.3.3.3.1.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_03				BBA_GAIN_03				LNA_GAIN_02				BBA_GAIN_02			
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	LNA_GAIN_01				BBA_GAIN_01				LNA_GAIN_00				BBA_GAIN_00			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.32.3 Fields

Field	Function
31-28 LNA_GAIN_03	LNA Gain 03 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24 BBA_GAIN_03	BBA Gain 03 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_02	LNA Gain 02 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_02	BBA Gain 02 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12	LNA Gain 01

Table continues on the next page...

Field	Function
LNA_GAIN_01	LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_01	BBA Gain 01 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_00	LNA Gain 00 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_00	BBA Gain 00 BBA GAIN.

45.3.3.3.1.33 AGC Gain Tables Step 07..04 (AGC_GAIN_TBL_07_04)

45.3.3.3.1.33.1 Offset

Register	Offset
AGC_GAIN_TBL_07_04	84h

45.3.3.3.1.33.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_07				BBA_GAIN_07				LNA_GAIN_06				BBA_GAIN_06			
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	LNA_GAIN_05				BBA_GAIN_05				LNA_GAIN_04				BBA_GAIN_04			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.33.3 Fields

Field	Function
31-28 LNA_GAIN_07	LNA Gain 07 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24 BBA_GAIN_07	BBA Gain 07 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_06	LNA Gain 06 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.

Table continues on the next page...

Field	Function
19-16 BBA_GAIN_06	BBA Gain 06 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_05	LNA Gain 05 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_05	BBA Gain 05 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_04	LNA Gain 04 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_04	BBA Gain 04 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

45.3.3.3.1.34 AGC Gain Tables Step 11..08 (AGC_GAIN_TBL_11_08)

45.3.3.3.1.34.1 Offset

Register	Offset
AGC_GAIN_TBL_11_08	88h

45.3.3.3.1.34.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_11				BBA_GAIN_11				LNA_GAIN_10				BBA_GAIN_10			
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	LNA_GAIN_09				BBA_GAIN_09				LNA_GAIN_08				BBA_GAIN_08			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.34.3 Fields

Field	Function
31-28 LNA_GAIN_11	LNA Gain 11 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24	BBA Gain 11

Table continues on the next page...

Field	Function
BBA_GAIN_11	BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_10	LNA Gain 10 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_10	BBA Gain 10 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_09	LNA Gain 09 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_09	BBA Gain 09 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_08	LNA Gain 08 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_08	BBA Gain 08 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

45.3.3.3.1.35 AGC Gain Tables Step 15..12 (AGC_GAIN_TBL_15_12)

45.3.3.3.1.35.1 Offset

Register	Offset
AGC_GAIN_TBL_15_12	8Ch

45.3.3.3.1.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_15				BBA_GAIN_15				LNA_GAIN_14				BBA_GAIN_14			
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	LNA_GAIN_13				BBA_GAIN_13				LNA_GAIN_12				BBA_GAIN_12			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.35.3 Fields

Field	Function
31-28	LNA Gain 15

Table continues on the next page...

Field	Function
LNA_GAIN_15	LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24 BBA_GAIN_15	BBA Gain 15 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_14	LNA Gain 14 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_14	BBA Gain 14 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_13	LNA Gain 13 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_13	BBA Gain 13 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_12	LNA Gain 12 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_12	BBA Gain 12 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

45.3.3.3.1.36 AGC Gain Tables Step 19..16 (AGC_GAIN_TBL_19_16)

45.3.3.3.1.36.1 Offset

Register	Offset
AGC_GAIN_TBL_19_16	90h

45.3.3.3.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_19				BBA_GAIN_19				LNA_GAIN_18				BBA_GAIN_18			
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	LNA_GAIN_17				BBA_GAIN_17				LNA_GAIN_16				BBA_GAIN_16			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.36.3 Fields

Field	Function
31-28 LNA_GAIN_19	LNA Gain 19 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24 BBA_GAIN_19	BBA Gain 193 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_18	LNA Gain 18 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_18	BBA Gain 18 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_17	LNA Gain 17 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_17	BBA Gain 17 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_16	LNA Gain 16 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_16	BBA Gain 16 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

45.3.3.3.1.37 AGC Gain Tables Step 23..20 (AGC_GAIN_TBL_23_20)

45.3.3.3.1.37.1 Offset

Register	Offset
AGC_GAIN_TBL_23_20	94h

45.3.3.3.1.37.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_23				BBA_GAIN_23				LNA_GAIN_22				BBA_GAIN_22			
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	LNA_GAIN_21				BBA_GAIN_21				LNA_GAIN_20				BBA_GAIN_20			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.37.3 Fields

Field	Function
31-28 LNA_GAIN_23	LNA Gain 23 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24 BBA_GAIN_23	BBA Gain 23 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_22	LNA Gain 22 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_22	BBA Gain 22 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_21	LNA Gain 21 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_21	BBA Gain 21 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_20	LNA Gain 20 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_20	BBA Gain 20 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

45.3.3.3.1.38 AGC Gain Tables Step 26..24 (AGC_GAIN_TBL_26_24)

45.3.3.3.1.38.1 Offset

Register	Offset
AGC_GAIN_TBL_26_24	98h

45.3.3.3.1.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								LNA_GAIN_26				BBA_GAIN_26			
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	LNA_GAIN_25				BBA_GAIN_25				LNA_GAIN_24				BBA_GAIN_24			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.38.3 Fields

Field	Function
31-24 —	Reserved.
23-20 LNA_GAIN_26	LNA Gain 26 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_26	BBA Gain 26 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_25	LNA Gain 25 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_25	BBA Gain 25 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_24	LNA Gain 24 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_24	BBA Gain 24 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

45.3.3.3.1.39 DCOC Offset (DCOC_OFFSET_0 - DCOC_OFFSET_26)

45.3.3.3.1.39.1 Offset

For a = 0 to 26:

Register	Offset
DCOC_OFFSET_a	A0h + (a × 4h)

45.3.3.3.1.39.2 Function

DCOC Offset Registers. These registers can only be accessed when the radio oscillator clock is active.

45.3.3.3.1.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DCOC_TZA_OFFSET_Q								DCOC_TZA_OFFSET_I							
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	0		DCOC_BBA_OFFSET_Q						0		DCOC_BBA_OFFSET_I					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.39.4 Fields

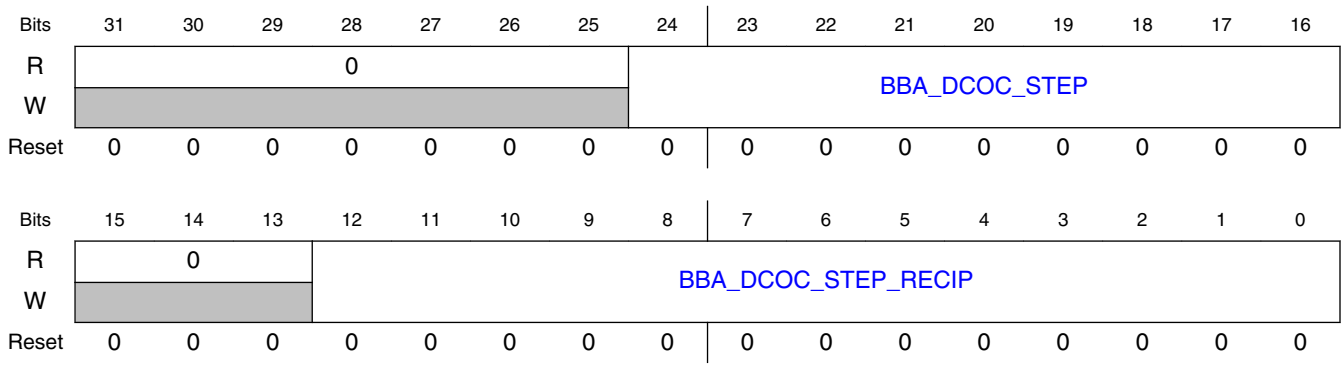
Field	Function
31-24 DCOC_TZA_OFFSET_Q	DCOC TZA Q-channel offset DCOC TZA Q-channel offset. When RX_DCOC_CAL_EN=1, this table is generated by the DCOC during calibration. When RX_DCOC_CAL_EN=0, this table may optionally be written by software.
23-16 DCOC_TZA_OFFSET_I	DCOC TZA I-channel offset DCOC TZA 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.
13-8 DCOC_BBA_OFFSET_Q	DCOC BBA Q-channel offset DCOC BBA 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.
7-6 —	Reserved.
5-0 DCOC_BBA_OFFSET_I	DCOC BBA I-channel offset DCOC BBA 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.

45.3.3.3.1.40 DCOC BBA DAC Step (DCOC_BBA_STEP)

45.3.3.3.1.40.1 Offset

Register	Offset
DCOC_BBA_STEP	10Ch

45.3.3.3.1.40.2 Diagram



45.3.3.3.1.40.3 Fields

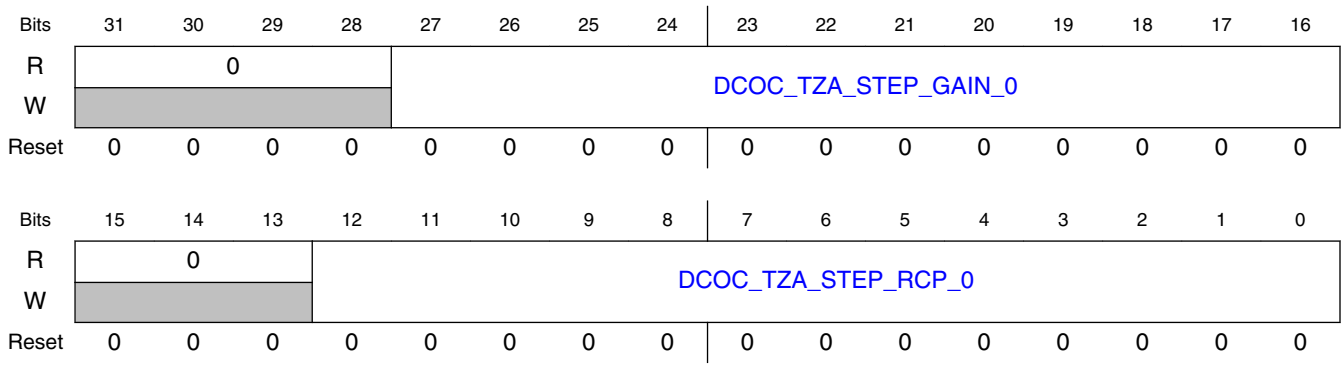
Field	Function
31-25 —	Reserved.
24-16 BBA_DCOC_STEP	DCOC BBA Step Size DCOC BBA Step Size (format: u6.3). This is the BBA DAC resolution in mV (1.2e3/32*63.44/120 = 19.83mV) times the ADC/decimator gain. This value is stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 BBA_DCOC_STEP_RECIP	DCOC BBA Reciprocal of Step Size DCOC BBA Reciprocal of Step Size (format: u.[00]13). This the reciprocal of the BBA DCOC STEP value. This value is stored as a 15 bit fraction (though only 13 bits are programmed).

45.3.3.3.1.41 DCOC TZA DAC Step 0 (DCOC_TZA_STEP_0)

45.3.3.3.1.41.1 Offset

Register	Offset
DCOC_TZA_STEP_0	110h

45.3.3.3.1.41.2 Diagram



45.3.3.3.1.41.3 Fields

Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_0	DCOC_TZA_STEP_GAIN_0 DCOC TZA Step Size 0 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 0. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_0	DCOC_TZA_STEP_RCP_0 DCOC TZA Reciprocal of Step Size 0, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_0. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.42 DCOC TZA DAC Step 1 (DCOC_TZA_STEP_1)

45.3.3.3.1.42.1 Offset

Register	Offset
DCOC_TZA_STEP_1	114h

45.3.3.3.1.42.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DCOC_TZA_STEP_GAIN_1											
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	0			DCOC_TZA_STEP_RCP_1												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.42.3 Fields

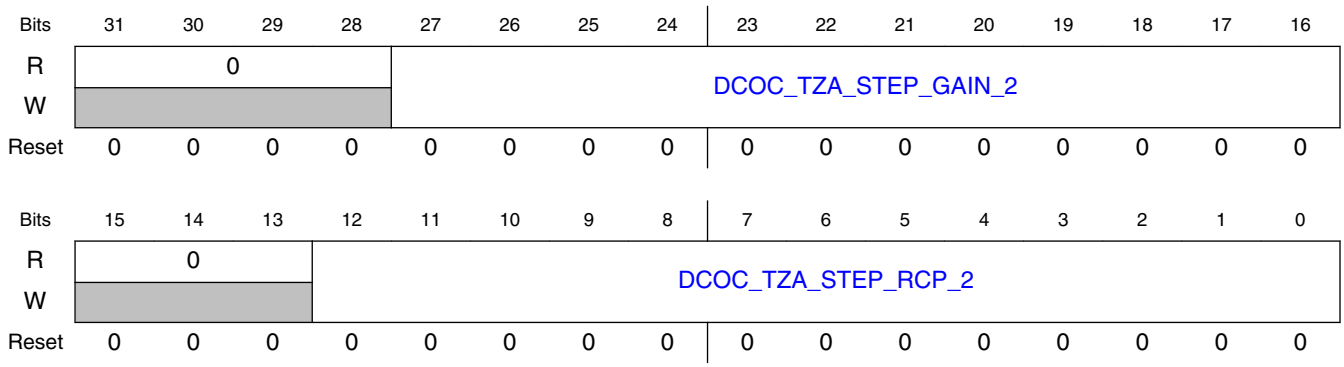
Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_1	DCOC_TZA_STEP_GAIN_1 DCOC TZA Step Size 1 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 1. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_1	DCOC_TZA_STEP_RCP_1 DCOC TZA Reciprocal of Step Size 1, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_1. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.43 DCOC TZA DAC Step 2 (DCOC_TZA_STEP_2)

45.3.3.3.1.43.1 Offset

Register	Offset
DCOC_TZA_STEP_2	118h

45.3.3.3.1.43.2 Diagram



45.3.3.3.1.43.3 Fields

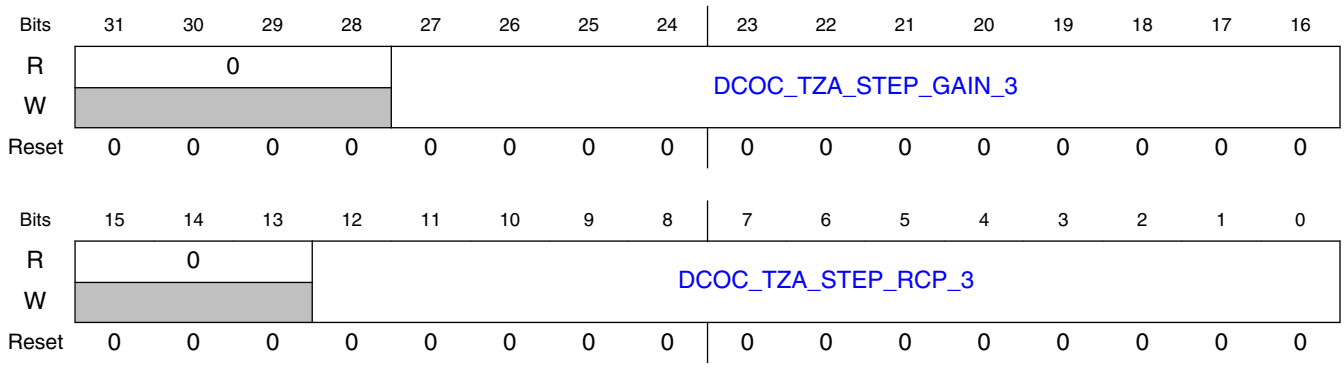
Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_2	DCOC_TZA_STEP_GAIN_2 DCOC TZA Step Size 2 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 2. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_2	DCOC_TZA_STEP_RCP_2 DCOC TZA Reciprocal of Step Size 2, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_2. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.44 DCOC TZA DAC Step 3 (DCOC_TZA_STEP_3)

45.3.3.3.1.44.1 Offset

Register	Offset
DCOC_TZA_STEP_3	11Ch

45.3.3.3.1.44.2 Diagram



45.3.3.3.1.44.3 Fields

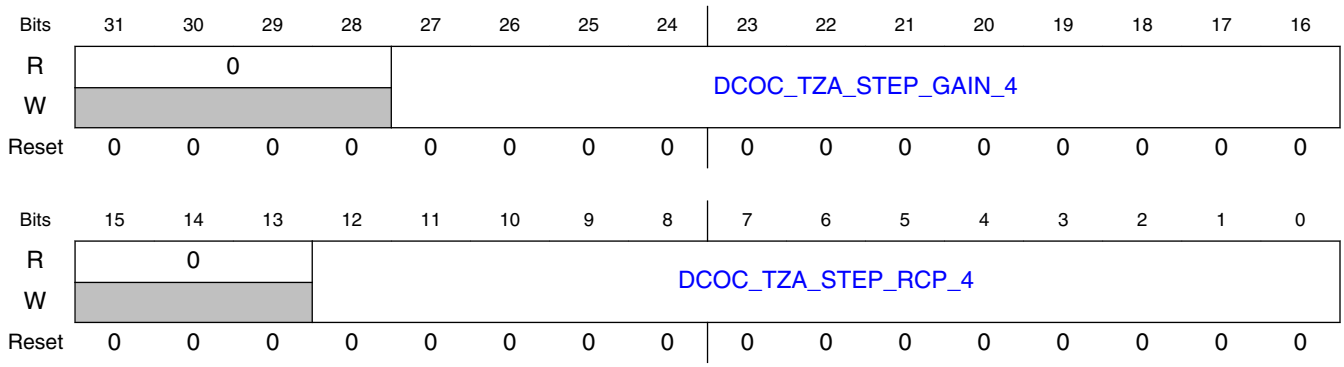
Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_3	DCOC_TZA_STEP_GAIN_3 DCOC TZA Step Size 3 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 3. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_3	DCOC_TZA_STEP_RCP_3 DCOC TZA Reciprocal of Step Size 3, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_3. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.45 DCOC TZA DAC Step 4 (DCOC_TZA_STEP_4)

45.3.3.3.1.45.1 Offset

Register	Offset
DCOC_TZA_STEP_4	120h

45.3.3.3.1.45.2 Diagram



45.3.3.3.1.45.3 Fields

Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_4	DCOC_TZA_STEP_GAIN_4 DCOC TZA Step Size 4 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 4. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_4	DCOC_TZA_STEP_RCP_4 DCOC TZA Reciprocal of Step Size 4, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_4. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.46 DCOC TZA DAC Step 5 (DCOC_TZA_STEP_5)

45.3.3.3.1.46.1 Offset

Register	Offset
DCOC_TZA_STEP_5	124h

45.3.3.3.1.46.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DCOC_TZA_STEP_GAIN_5											
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	0			DCOC_TZA_STEP_RCP_5												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.46.3 Fields

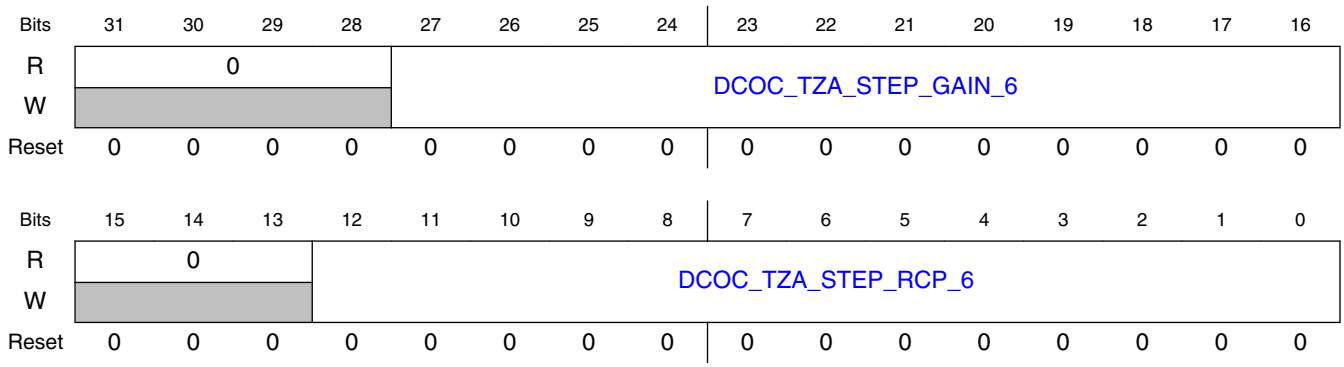
Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_5	DCOC_TZA_STEP_GAIN_5 DCOC TZA Step Size 5 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 5. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_5	DCOC_TZA_STEP_RCP_5 DCOC TZA Reciprocal of Step Size 5, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_5. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.47 DCOC TZA DAC Step 6 (DCOC_TZA_STEP_6)

45.3.3.3.1.47.1 Offset

Register	Offset
DCOC_TZA_STEP_6	128h

45.3.3.3.1.47.2 Diagram



45.3.3.3.1.47.3 Fields

Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_6	DCOC_TZA_STEP_GAIN_6 DCOC TZA Step Size 6 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 6. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_6	DCOC_TZA_STEP_RCP_6 DCOC TZA Reciprocal of Step Size 6, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_6. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.48 DCOC TZA DAC Step 7 (DCOC_TZA_STEP_7)

45.3.3.3.1.48.1 Offset

Register	Offset
DCOC_TZA_STEP_7	12Ch

45.3.3.3.1.48.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DCOC_TZA_STEP_GAIN_7												
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	0			DCOC_TZA_STEP_RCP_7												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.48.3 Fields

Field	Function
31-29 —	Reserved.
28-16 DCOC_TZA_STEP_GAIN_7	DCOC_TZA_STEP_GAIN_7 DCOC TZA Step Size 7 with gain (format u10.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 7. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_7	DCOC_TZA_STEP_RCP_7 DCOC TZA Reciprocal of Step Size 7, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_7. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.49 DCOC TZA DAC Step 5 (DCOC_TZA_STEP_8)

45.3.3.3.1.49.1 Offset

Register	Offset
DCOC_TZA_STEP_8	130h

45.3.3.3.1.49.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DCOC_TZA_STEP_GAIN_8												
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	0			DCOC_TZA_STEP_RCP_8												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.49.3 Fields

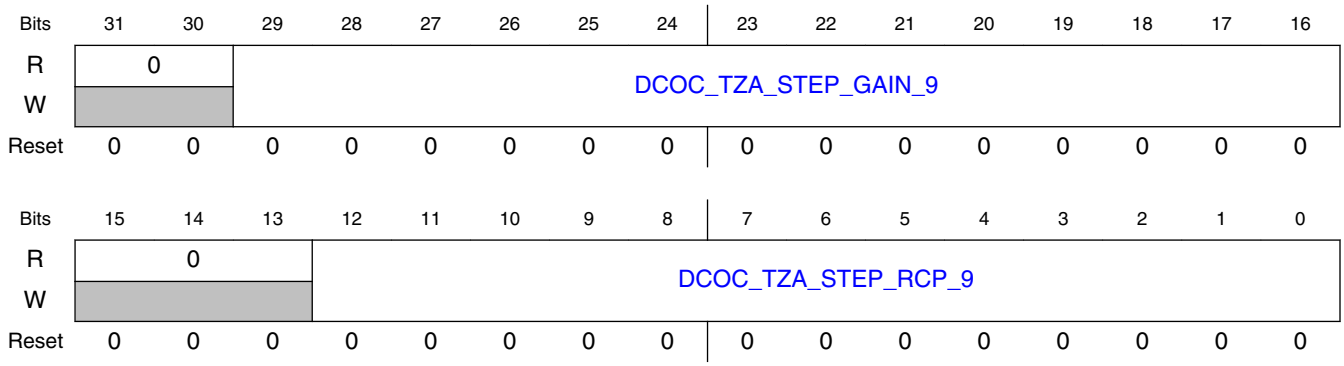
Field	Function
31-29 —	Reserved.
28-16 DCOC_TZA_STEP_GAIN_8	DCOC_TZA_STEP_GAIN_8 DCOC TZA Step Size 8 with gain (format u10.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 8. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_8	DCOC_TZA_STEP_RCP_8 DCOC TZA Reciprocal of Step Size 8, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_8. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.50 DCOC TZA DAC Step 9 (DCOC_TZA_STEP_9)

45.3.3.3.1.50.1 Offset

Register	Offset
DCOC_TZA_STEP_9	134h

45.3.3.3.1.50.2 Diagram



45.3.3.3.1.50.3 Fields

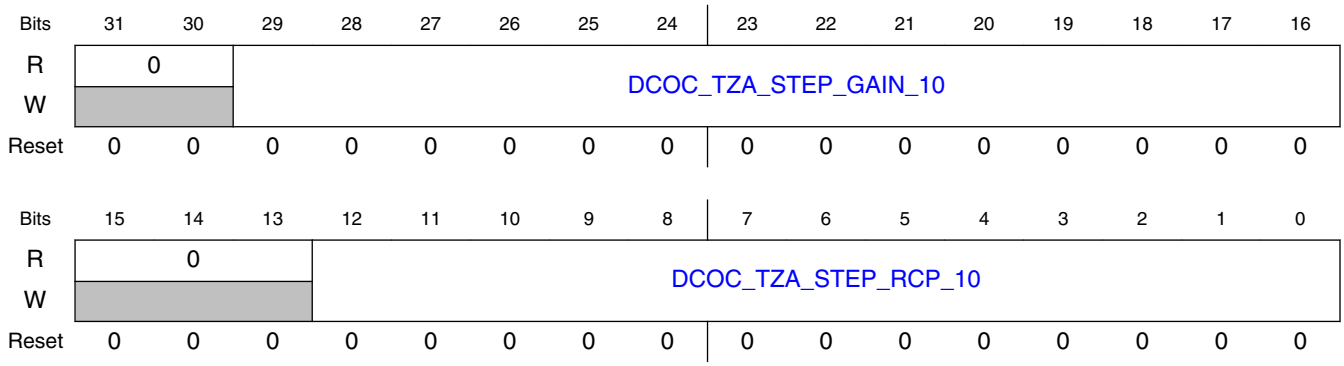
Field	Function
31-30 —	Reserved.
29-16 DCOC_TZA_STEP_GAIN_9	DCOC_TZA_STEP_GAIN_9 DCOC TZA Step Size 9 with gain (format u11.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 9. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_9	DCOC_TZA_STEP_RCP_9 DCOC TZA Reciprocal of Step Size 9, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_9. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.51 DCOC TZA DAC Step 10 (DCOC_TZA_STEP_10)

45.3.3.3.1.51.1 Offset

Register	Offset
DCOC_TZA_STEP_10	138h

45.3.3.3.1.51.2 Diagram



45.3.3.3.1.51.3 Fields

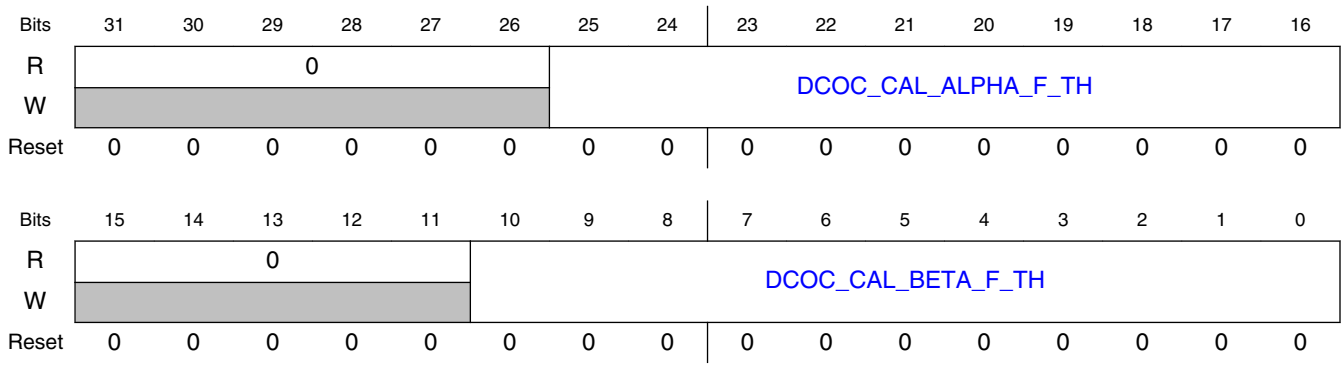
Field	Function
31-30 —	Reserved.
29-16 DCOC_TZA_STEP_GAIN_10	DCOC_TZA_STEP_GAIN_10 DCOC TZA Step Size 10 with gain (format u11.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 10. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_10	DCOC_TZA_STEP_RCP_10 DCOC TZA Reciprocal of Step Size 10, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_10. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

45.3.3.3.1.52 DCOC Calibration Fail Thresholds (DCOC_CAL_FAIL_TH)

45.3.3.3.1.52.1 Offset

Register	Offset
DCOC_CAL_FAIL_TH	160h

45.3.3.3.1.52.2 Diagram



45.3.3.3.1.52.3 Fields

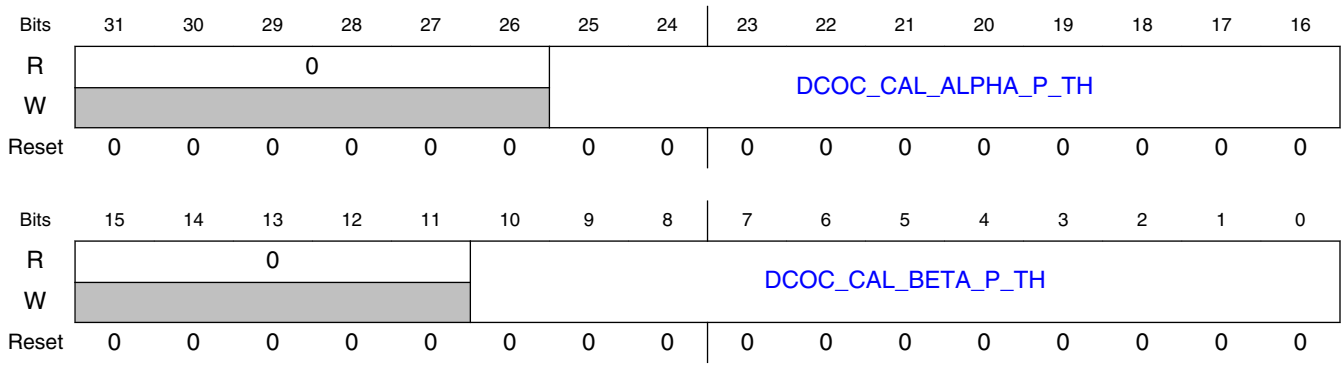
Field	Function
31-26 —	Reserved.
25-16 DCOC_CAL_ALPHA_F_TH	DCOC Calibration Alpha Fail Threshold If DCOC_CTRL_0[DCOC_CHECK_EN] is set, the absolute value of the alpha-hat I and Q estimates are compared gainst this fail threshold. If any of the estimates are greater than the fail threshold, the calibration is considered a failure. Format is u4.6.
15-11 —	Reserved.
10-0 DCOC_CAL_BETA_F_TH	DCOC Calibration Beta Fail Threshold If DCOC_CTRL_0[DCOC_CHECK_EN] is set, the absolute value of the beta-hat I and Q estimates are compared against this fail threshold. If any of the estimates are greater than the fail threshold, the calibration is considered a failure. Format is u11.

45.3.3.3.1.53 DCOC Calibration Pass Thresholds (DCOC_CAL_PASS_TH)

45.3.3.3.1.53.1 Offset

Register	Offset
DCOC_CAL_PASS_TH	164h

45.3.3.3.1.53.2 Diagram



45.3.3.3.1.53.3 Fields

Field	Function
31-26 —	Reserved.
25-16 DCOC_CAL_ALPHA_P_TH	DCOC Calibration Alpha Pass Threshold If DCOC_CTRL_0[DCOC_CHECK_EN] is set, the absolute value of the alpha-hat I and Q estimates are compared against this pass threshold. If all of the estimates are less than the pass threshold, the calibration is considered as passing. Format is u4.6.
15-11 —	Reserved.
10-0 DCOC_CAL_BETA_P_TH	DCOC Calibration Beta Pass Threshold If DCOC_CTRL_0[DCOC_CHECK_EN] is set, the absolute value of the beta-hat I and Q estimates are compared against this pass threshold. If all of the estimates are less than the pass threshold, the calibration is considered as passing. Format is u11.

45.3.3.3.1.54 DCOC Calibration Alpha (DCOC_CAL_ALPHA)

45.3.3.3.1.54.1 Offset

Register	Offset
DCOC_CAL_ALPHA	168h

45.3.3.3.1.54.2 Diagram

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

45.3.3.3.1.54.3 Fields

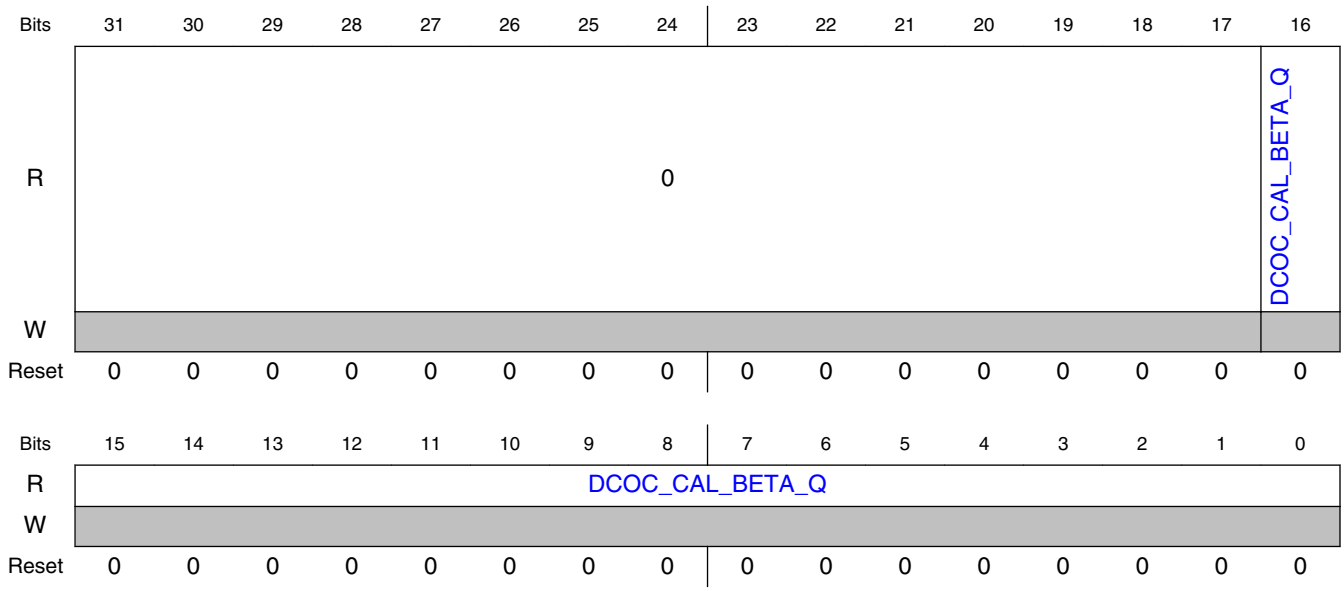
Field	Function
31-27 —	Reserved.
26-16 DCOC_CAL_ALPHA_Q	DCOC_CAL_ALPHA_Q DCOC Calibration Q-channel ALPHA. This read-only value represents the Q channel estimate of the ALPHA DC component calculated in DCOC calibration. Format s4.6. This is provided for debug/ characterization purposes.
15-11 —	Reserved.
10-0 DCOC_CAL_ALPHA_I	DCOC Calibration I-channel ALPHA constant DCOC Calibration I-channel ALPHA. This read-only value represents the I channel estimate of the ALPHA DC component calculated in DCOC calibration. Format s4.6. This is provided for debug/ characterization purposes.

45.3.3.3.1.55 DCOC Calibration Beta Q (DCOC_CAL_BETA_Q)

45.3.3.3.1.55.1 Offset

Register	Offset
DCOC_CAL_BETA_Q	16Ch

45.3.3.3.1.55.2 Diagram



45.3.3.3.1.55.3 Fields

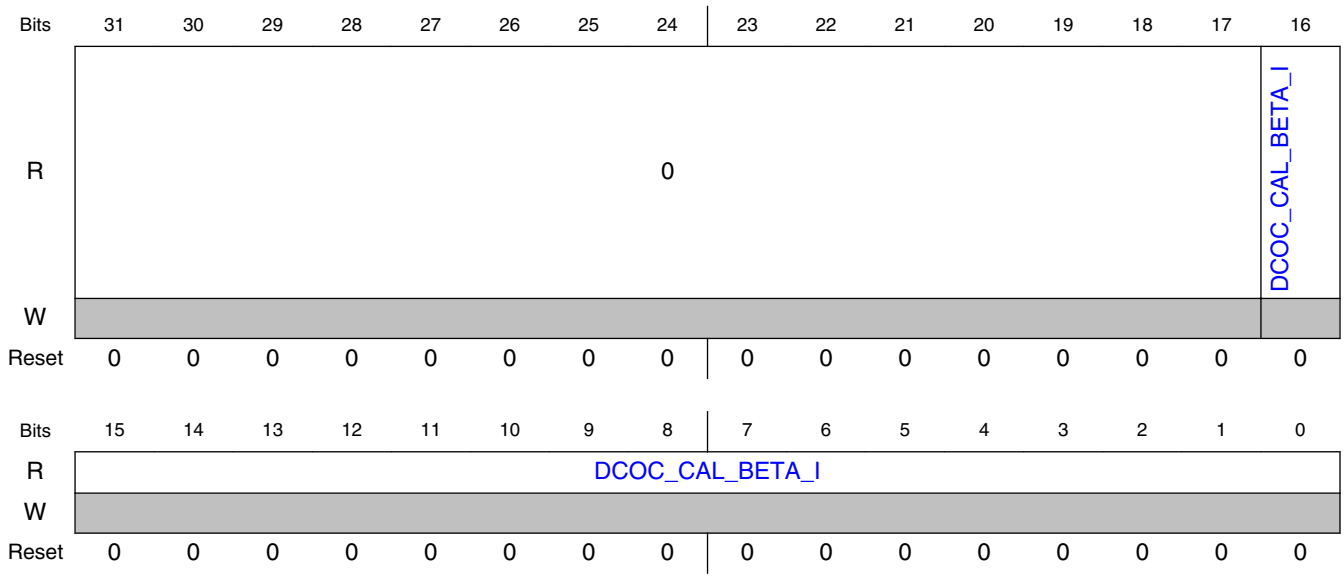
Field	Function
31-17 —	Reserved.
16-0 DCOC_CAL_BETA_Q	DCOC_CAL_BETA_Q DCOC Calibration Q-channel BETA. This read-only value represents the Q channel estimate of the BETA DC component calculated in DCOC calibration. Format s11.5. This is provided for debug/characterization purposes.

45.3.3.3.1.56 DCOC Calibration Beta I (DCOC_CAL_BETA_I)

45.3.3.3.1.56.1 Offset

Register	Offset
DCOC_CAL_BETA_I	170h

45.3.3.3.1.56.2 Diagram



45.3.3.3.1.56.3 Fields

Field	Function
31-17 —	Reserved.
16-0 DCOC_CAL_BETA_I	DCOC_CAL_BETA_I DCOC Calibration I-channel BETA. This read-only value represents the I channel estimate of the BETA DC component calculated in DCOC calibration. Format s11.5. This is provided for debug/characterization purposes.

45.3.3.3.1.57 DCOC Calibration Gamma (DCOC_CAL_GAMMA)

45.3.3.3.1.57.1 Offset

Register	Offset
DCOC_CAL_GAMMA	174h

45.3.3.3.1.57.2 Diagram

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

45.3.3.3.1.57.3 Fields

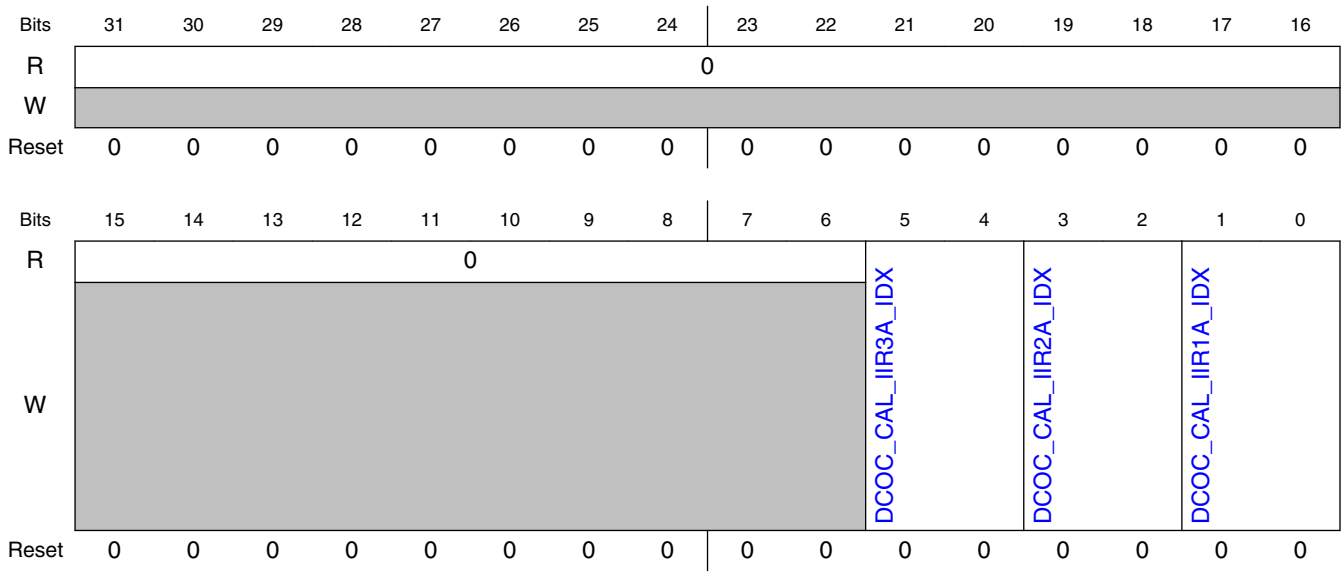
Field	Function
31-16 DCOC_CAL_GAMMA_Q	DCOC_CAL_GAMMA_Q DCOC Calibration Q-channel GAMMA. This read-only value represents the Q channel estimate of the GAMMA DC component calculated in DCOC calibration. Format s11.4. This is provided for debug/ characterization purposes.
15-0 DCOC_CAL_GAMMA_I	DCOC_CAL_GAMMA_I DCOC Calibration I-channel GAMMA. This read-only value represents the I channel estimate of the GAMMA DC component calculated in DCOC calibration. Format s11.4. This is provided for debug/ characterization purposes.

45.3.3.3.1.58 DCOC Calibration IIR (DCOC_CAL_IIR)

45.3.3.3.1.58.1 Offset

Register	Offset
DCOC_CAL_IIR	178h

45.3.3.3.1.58.2 Diagram



45.3.3.3.1.58.3 Fields

Field	Function
31-6 —	Reserved.
5-4 DCOC_CAL_IIR 3A_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. 00b - 1/4 01b - 1/8 10b - 1/16 11b - 1/32
3-2 DCOC_CAL_IIR 2A_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. 00b - 1/1 01b - 1/4 10b - 1/8 11b - 1/16
1-0 DCOC_CAL_IIR 1A_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. 00b - 1/1 01b - 1/4 10b - 1/8 11b - 1/16

45.3.3.3.1.59 DCOC Calibration Result (DCOC_CAL1 - DCOC_CAL3)

45.3.3.3.1.59.1 Offset

For a = 1 to 3:

Register	Offset
DCOC_CALa	17Ch + (a × 4h)

45.3.3.3.1.59.2 Function

Result of one of the calibration iterations.

45.3.3.3.1.59.3 Diagram

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

45.3.3.3.1.59.4 Fields

Field	Function
31-28 —	Reserved.
27-16 DCOC_CAL_RES_Q	DCOC Calibration Result - Q Channel Q channel DCOC calibration result. This value represents the DCOC's Q channel DC estimate for the nth calibration gain setting. Format s11. This is provided for debug/chacterization purposes.
15-12 —	Reserved.
11-0 DCOC_CAL_RES_I	DCOC Calibration Result - I Channel I channel DCOC calibration result. This value represents the DCOC's I channel DC estimate for the nth calibration gain setting. Format s11. This is provided for debug/chacterization purposes.

45.3.3.3.1.60 RX_DIG CCA ED LQI Control Register 0 (CCA_ED_LQI_CTRL_0)

45.3.3.3.1.60.1 Offset

Register	Offset
CCA_ED_LQI_CTRL_0	190h

45.3.3.3.1.60.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		SNR_ADJ						LQI_CNTR							
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	CORR_CNTR_THRESH								LQI_CORR_THRESH							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.60.3 Fields

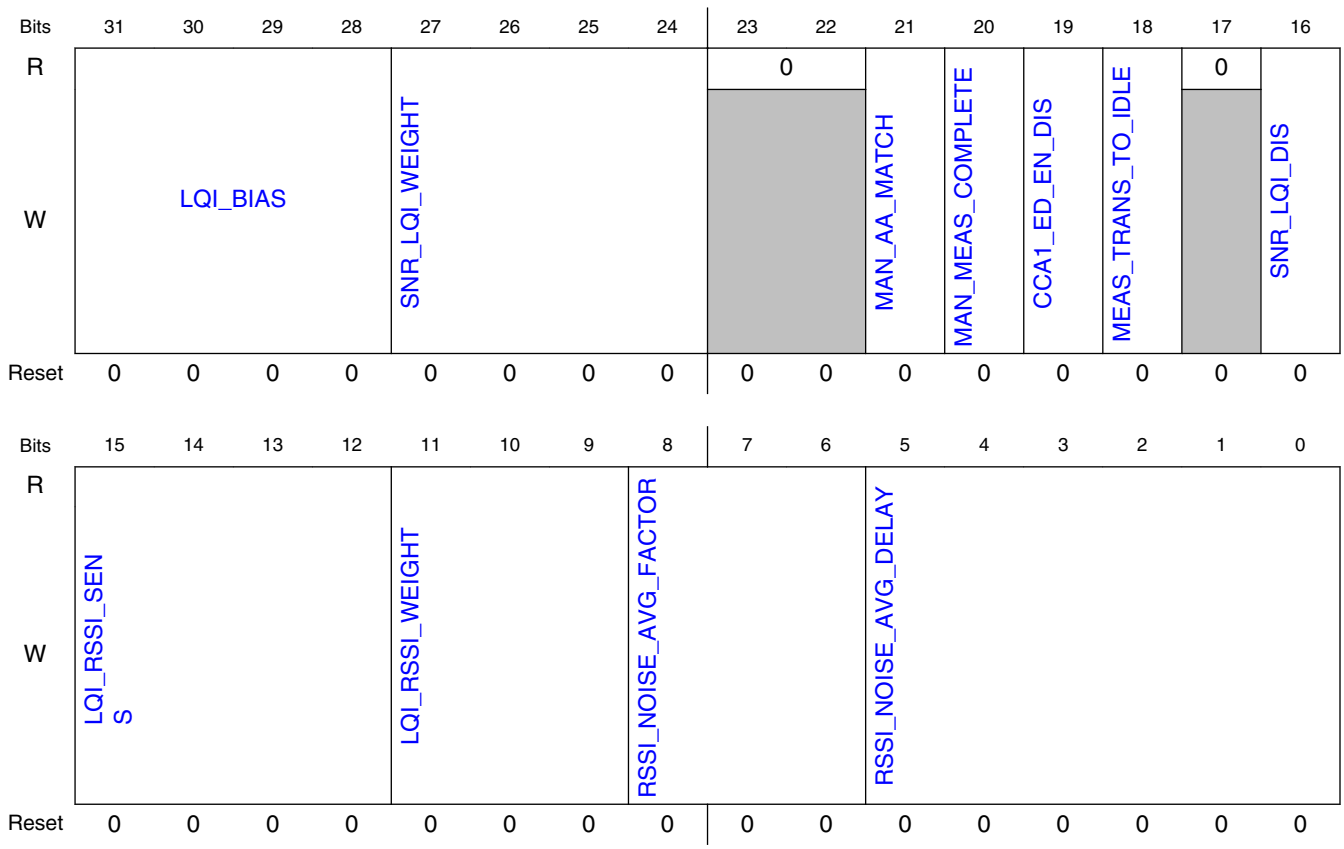
Field	Function
31-30 —	Reserved.
29-24 SNR_ADJ	SNR calculation adjustment Signed value with quarter dB resolution.
23-16 LQI_CNTR	LQI Counter Duration of LQI in uS.
15-8 CORR_CNTR_THRESH	Correlation Count Threshold Threshold used to compare the counted correlation magnitudes exceeding LQI_CORR_THRESH.
7-0 LQI_CORR_THRESH	LQI Correlation Threshold Threshold used to compare correlation magnitude values from the PHY

45.3.3.3.1.61 RX_DIG CCA ED LQI Control Register 1 (CCA_ED_LQI_CTRL_1)

45.3.3.3.1.61.1 Offset

Register	Offset
CCA_ED_LQI_CTRL_1	194h

45.3.3.3.1.61.2 Diagram



45.3.3.3.1.61.3 Fields

Field	Function
31-28 LQI_BIAS	LQI Bias. Bias used for LQI calculation. Applied bias = -36 + 2*LQI_BIAS.
27-24 SNR_LQI_WEIGHT	SNR LQI Weight SNR weight used for LQI calculation 0000b - 0.0 0001b - 1.0 0010b - 1.125 0011b - 1.25 0100b - 1.375 0101b - 1.5

Table continues on the next page...

Field	Function
	0110b - 1.625 0111b - 1.75 1000b - 1.875 1001b - 2.0 1010b - 2.125 1011b - 2.25 1100b - 2.375 1101b - 2.5 1110b - 2.625 1111b - 2.75
23-22 ——	Reserved.
21 MAN_AA_MATCH	Manual AA Match When set, this causes an AA match condition. Intended to be used only for debug. 0b - Normal operation 1b - Manually asserts the AA match signal for the RX_DIG CCA/ED/LQI and AGC blocks. Intended to be used only for debug.
20 MAN_MEAS_COMPLETE	Manual measurement complete 0b - Normal operation 1b - Manually asserts the measurement complete signal for the RX_DIG CCA/ED/LQI blocks. Intended to be used only for debug.
19 CCA1_ED_EN_DIS	CCA1_ED_EN Disable 0b - Normal operation 1b - CCA1_ED_EN input is disabled
18 MEAS_TRANS_TO_IDLE	Measurement Transition to IDLE Establishes the state machine transition following an LQI or CCA1/ED measurement 0b - Module transitions to RSSI state 1b - Module transitions to IDLE state
17 ——	Reserved.
16 SNR_LQI_DIS	SNR LQI Disable 0b - Normal operation. 1b - The RX_DIG CCA/ED/LQI block ignores the AA match input which starts an LQI measurement.
15-12 LQI_RSSI_SENS	LQI RSSI Sensitivity Unsigned integer used for calculation of LQI sensitivity. Sensitivity = -103 + LQI_RSSI_SENS.
11-9 LQI_RSSI_WEIGHT	LQI RSSI Weight RSSI weight used for LQI calculation 000b - 2.0 001b - 2.125 010b - 2.25 011b - 2.375 100b - 2.5 101b - 2.625 110b - 2.75 111b - 2.875
8-6	RSSI Noise Averaging Factor Factor used for RSSI and SNR averaging.

Table continues on the next page...

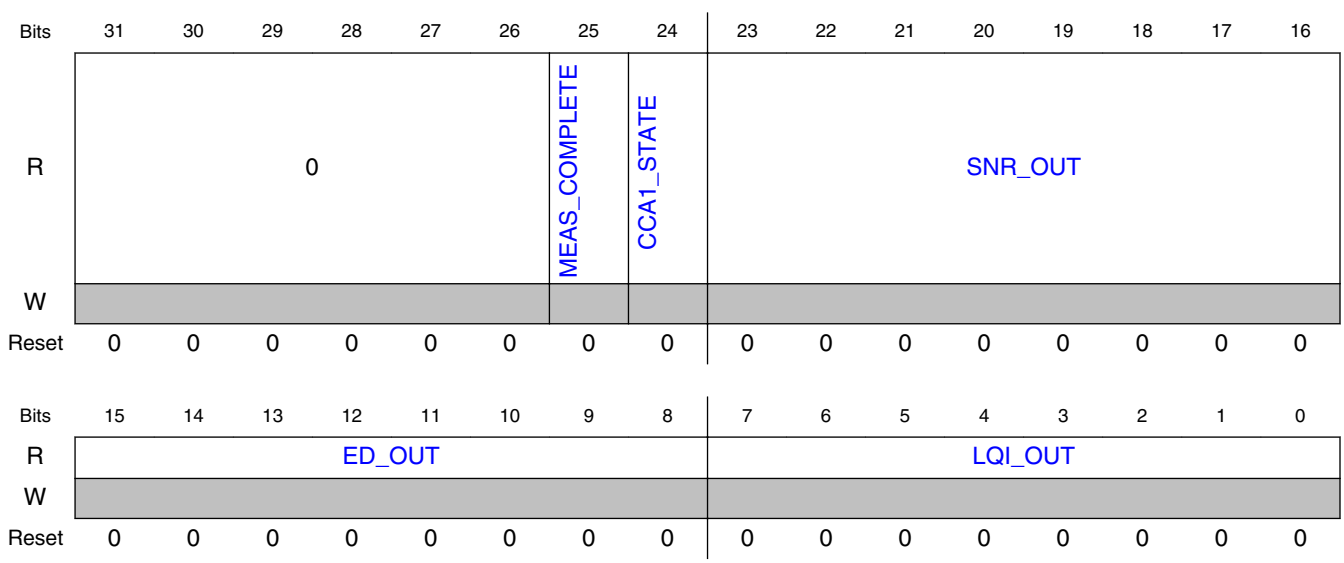
Field	Function
RSSI_NOISE_AVG_FACTOR	000b - 1 001b - 64 010b - 70 011b - 128 100b - 139 101b - 256 110b - 277 111b - 512
5-0 RSSI_NOISE_AVG_DELAY	RSSI Noise Averaging Delay Programmable delay used to delay the enable of averagers "RSSI Averager" and "Noise Averager" after a cca1_ed_trig assertion (in wideband mode) or after a aa_sfd_matched assertion (in narrowband mode). The delay is expressed in ticks of frequenc Fs/N, where Fs is the ADC decimator output sampling frequency in wideband mode or the base-band sampling frequency in the narrowband mode.

45.3.3.3.1.62 RX_DIG CCA ED LQI Status Register 0 (CCA_ED_LQI_STAT_0)

45.3.3.3.1.62.1 Offset

Register	Offset
CCA_ED_LQI_STAT_0	198h

45.3.3.3.1.62.2 Diagram



45.3.3.3.1.62.3 Fields

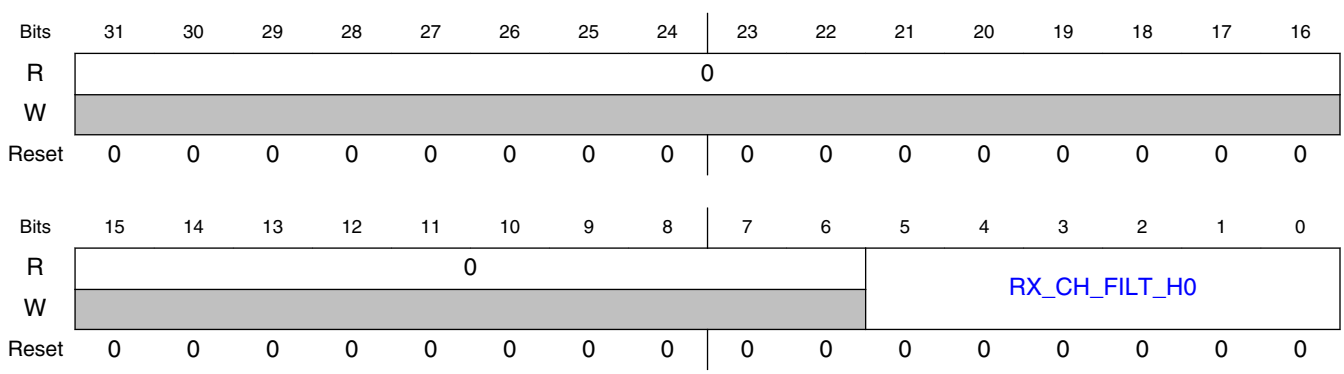
Field	Function
31-26 —	Reserved.
25 MEAS_COMPLETE	Measurement Complete Set upon measurement complete
24 CCA1_STATE	CCA1 State Reflects the state of CCA1 channel state
23-16 SNR_OUT	SNR output Reflects the SNR measurement value.
15-8 ED_OUT	ED output Reflects the Energy Detect (DC) measurement value.
7-0 LQI_OUT	LQI output Reflects the LQI measurement value.

45.3.3.3.1.63 Receive Channel Filter Coefficient 0 (RX_CHF_COEF_0)

45.3.3.3.1.63.1 Offset

Register	Offset
RX_CHF_COEF_0	1A0h

45.3.3.3.1.63.2 Diagram



45.3.3.3.1.63.3 Fields

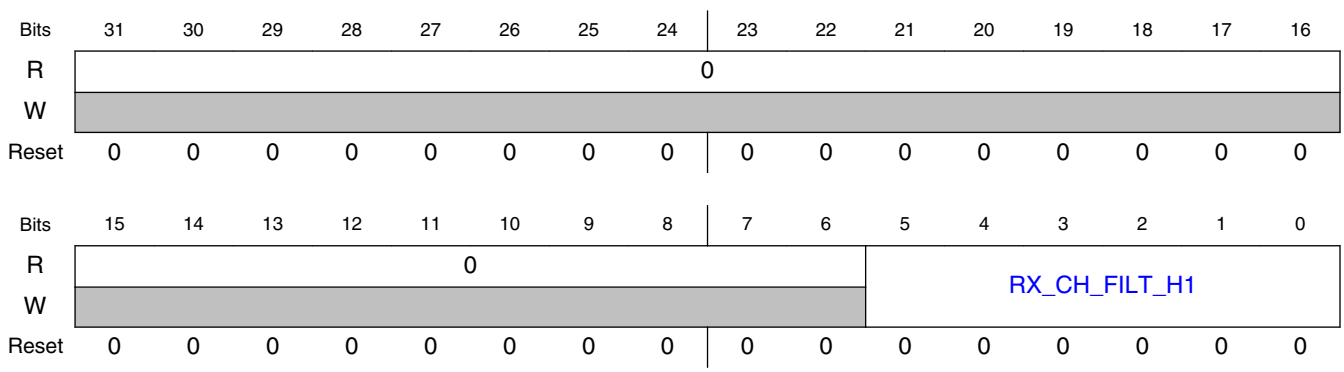
Field	Function
31-6 —	Reserved.
5-0 RX_CH_FILT_H0	RX Channel Filter Coefficient 0 RX Channel Filter Coefficient 0, 6-bit signed fractional

45.3.3.3.1.64 Receive Channel Filter Coefficient 1 (RX_CHF_COEF_1)

45.3.3.3.1.64.1 Offset

Register	Offset
RX_CHF_COEF_1	1A4h

45.3.3.3.1.64.2 Diagram



45.3.3.3.1.64.3 Fields

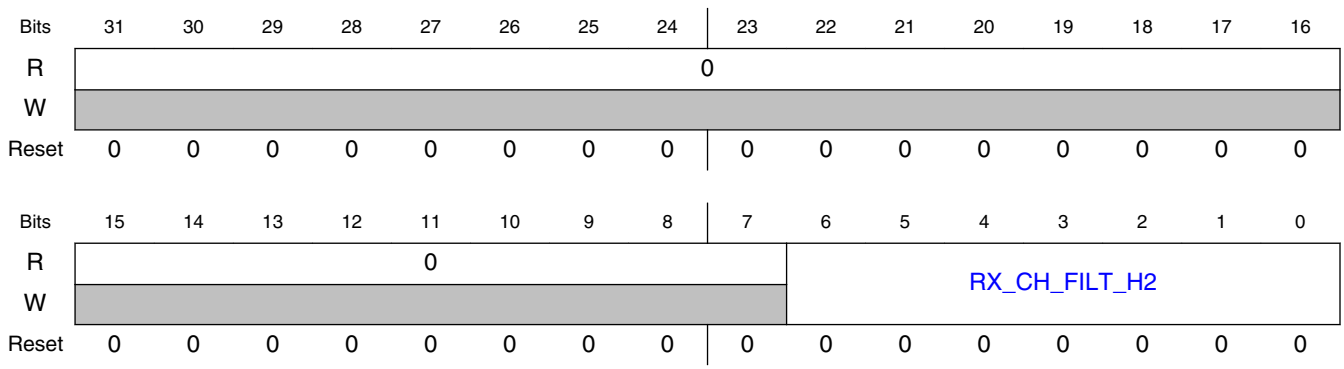
Field	Function
31-6 —	Reserved.
5-0 RX_CH_FILT_H1	RX Channel Filter Coefficient 1 RX Channel Filter Coefficient 1, 6-bit signed fractional

45.3.3.3.1.65 Receive Channel Filter Coefficient 2 (RX_CHF_COEF_2)

45.3.3.3.1.65.1 Offset

Register	Offset
RX_CHF_COEF_2	1A8h

45.3.3.3.1.65.2 Diagram



45.3.3.3.1.65.3 Fields

Field	Function
31-7 —	Reserved.
6-0 RX_CH_FILT_H2	RX Channel Filter Coefficient 2 RX Channel Filter Coefficient 2, 7-bit signed fractional

45.3.3.3.1.66 Receive Channel Filter Coefficient 3 (RX_CHF_COEF_3)

45.3.3.3.1.66.1 Offset

Register	Offset
RX_CHF_COEF_3	1ACh

45.3.3.3.1.66.2 Diagram

Bits	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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RX_CH_FILT_H3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.66.3 Fields

Field	Function
31-7 —	Reserved.
6-0 RX_CH_FILT_H3	RX Channel Filter Coefficient 3 RX Channel Filter Coefficient 3, 7-bit signed fractional

45.3.3.3.1.67 Receive Channel Filter Coefficient 4 (RX_CHF_COEF_4)

45.3.3.3.1.67.1 Offset

Register	Offset
RX_CHF_COEF_4	1B0h

45.3.3.3.1.67.2 Diagram

Bits	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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RX_CH_FILT_H4							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.67.3 Fields

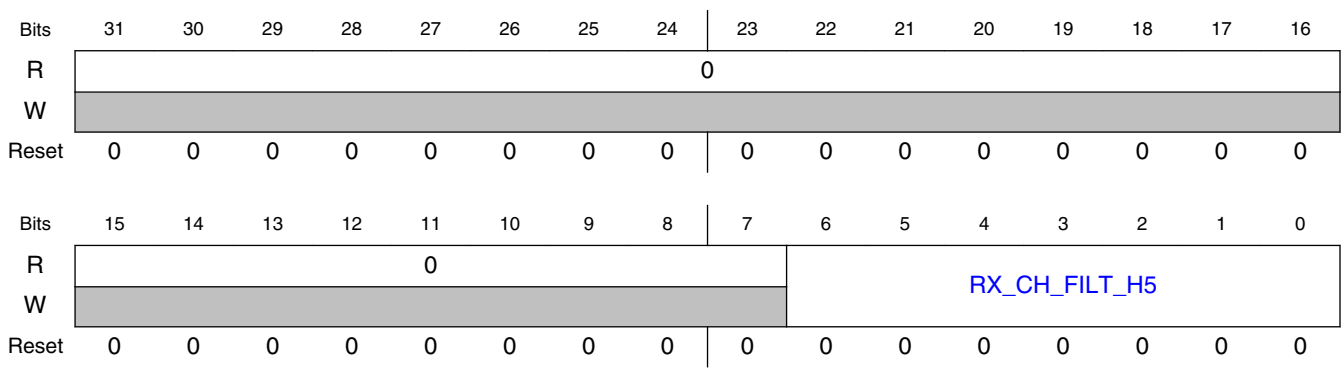
Field	Function
31-7 —	Reserved.
6-0 RX_CH_FILT_H4	RX Channel Filter Coefficient 4 RX Channel Filter Coefficient 4, 7-bit signed fractional

45.3.3.3.1.68 Receive Channel Filter Coefficient 5 (RX_CHF_COEF_5)

45.3.3.3.1.68.1 Offset

Register	Offset
RX_CHF_COEF_5	1B4h

45.3.3.3.1.68.2 Diagram



45.3.3.3.1.68.3 Fields

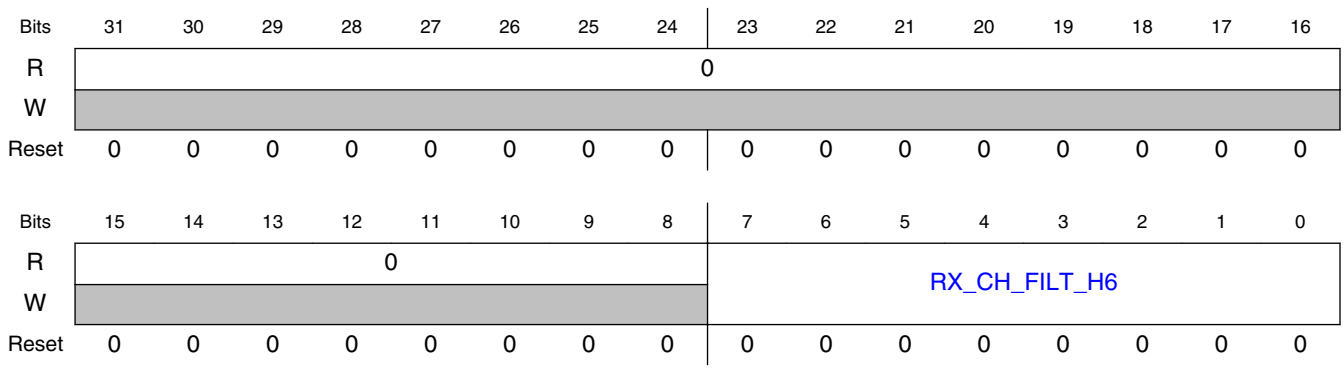
Field	Function
31-7 —	Reserved.
6-0 RX_CH_FILT_H5	RX Channel Filter Coefficient 5 RX Channel Filter Coefficient 5, 7-bit signed fractional

45.3.3.3.1.69 Receive Channel Filter Coefficient 6 (RX_CHF_COEF_6)

45.3.3.3.1.69.1 Offset

Register	Offset
RX_CHF_COEF_6	1B8h

45.3.3.3.1.69.2 Diagram



45.3.3.3.1.69.3 Fields

Field	Function
31-8 —	Reserved.
7-0 RX_CH_FILT_H6	RX Channel Filter Coefficient 6 RX Channel Filter Coefficient 6, 8-bit signed fractional

45.3.3.3.1.70 Receive Channel Filter Coefficient 7 (RX_CHF_COEF_7)

45.3.3.3.1.70.1 Offset

Register	Offset
RX_CHF_COEF_7	1BCh

45.3.3.3.1.70.2 Diagram

Bits	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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RX_CH_FILT_H7							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.70.3 Fields

Field	Function
31-8 —	Reserved.
7-0 RX_CH_FILT_H7	RX Channel Filter Coefficient 7 RX Channel Filter Coefficient 7, 8-bit signed fractional

45.3.3.3.1.71 Receive Channel Filter Coefficient 8 (RX_CHF_COEF_8)

45.3.3.3.1.71.1 Offset

Register	Offset
RX_CHF_COEF_8	1C0h

45.3.3.3.1.71.2 Diagram

Bits	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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RX_CH_FILT_H8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.71.3 Fields

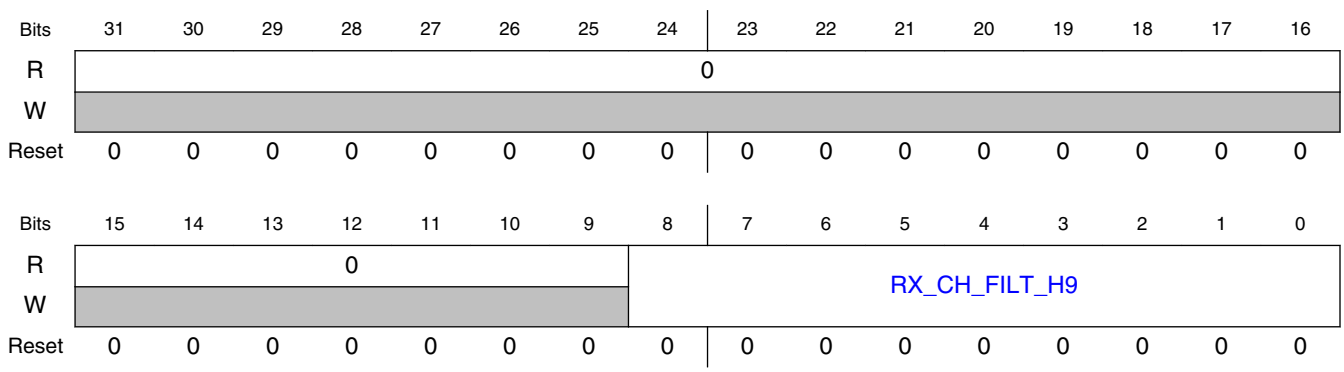
Field	Function
31-9 —	Reserved.
8-0 RX_CH_FILT_H8	RX Channel Filter Coefficient 8 RX Channel Filter Coefficient 8, 9-bit signed fractional

45.3.3.3.1.72 Receive Channel Filter Coefficient 9 (RX_CHF_COEF_9)

45.3.3.3.1.72.1 Offset

Register	Offset
RX_CHF_COEF_9	1C4h

45.3.3.3.1.72.2 Diagram



45.3.3.3.1.72.3 Fields

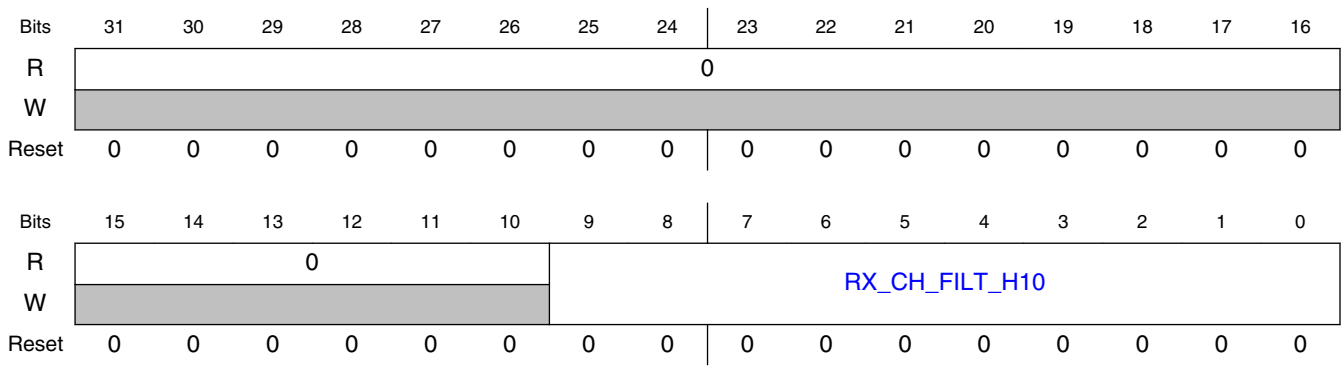
Field	Function
31-9 —	Reserved.
8-0 RX_CH_FILT_H9	RX Channel Filter Coefficient 9 RX Channel Filter Coefficient 9, 9-bit signed fractional

45.3.3.3.1.73 Receive Channel Filter Coefficient 10 (RX_CHF_COEF_10)

45.3.3.3.1.73.1 Offset

Register	Offset
RX_CHF_COEF_10	1C8h

45.3.3.3.1.73.2 Diagram



45.3.3.3.1.73.3 Fields

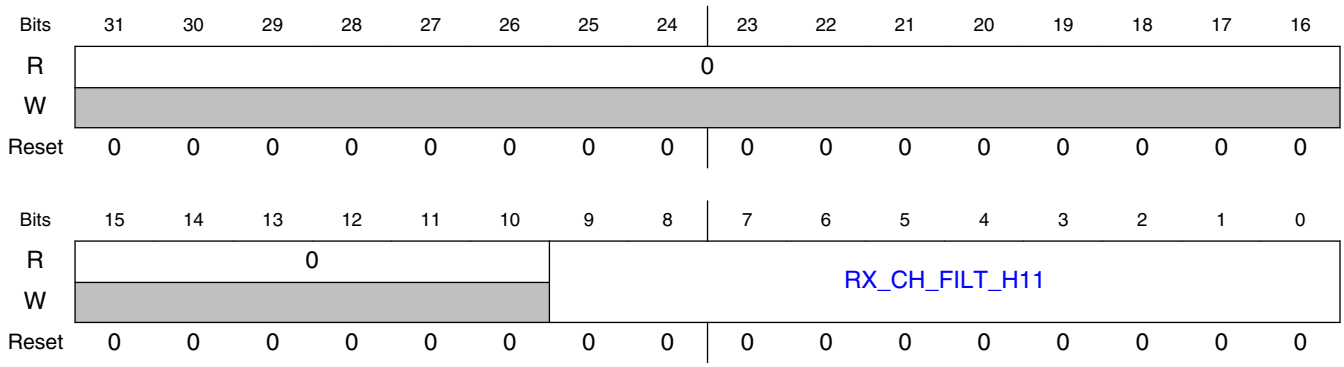
Field	Function
31-10 —	Reserved.
9-0 RX_CH_FILT_H10	RX Channel Filter Coefficient 10 RX Channel Filter Coefficient 10, 10-bit signed fractional

45.3.3.3.1.74 Receive Channel Filter Coefficient 11 (RX_CHF_COEF_11)

45.3.3.3.1.74.1 Offset

Register	Offset
RX_CHF_COEF_11	1CCh

45.3.3.3.1.74.2 Diagram



45.3.3.3.1.74.3 Fields

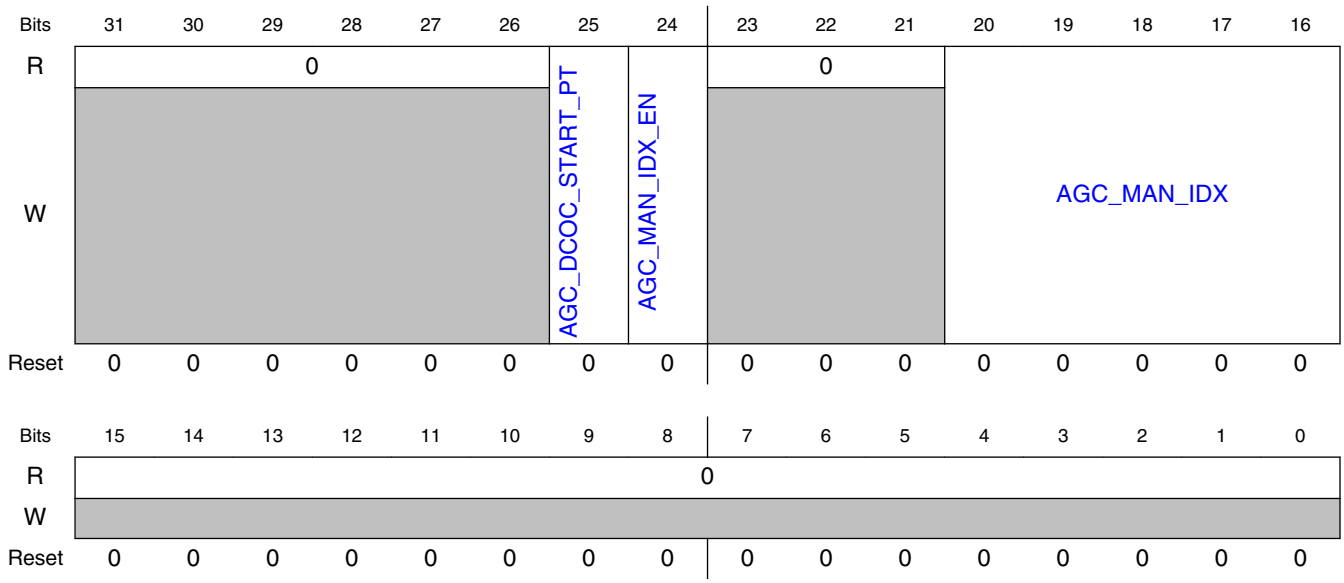
Field	Function
31-10 —	Reserved.
9-0 RX_CH_FILT_H11	RX Channel Filter Coefficient 11 RX Channel Filter Coefficient 11, 10-bit signed fractional

45.3.3.3.1.75 AGC Manual AGC Index (AGC_MAN_AGC_IDX)

45.3.3.3.1.75.1 Offset

Register	Offset
AGC_MAN_AGC_IDX	1D0h

45.3.3.3.1.75.2 Diagram



45.3.3.3.1.75.3 Fields

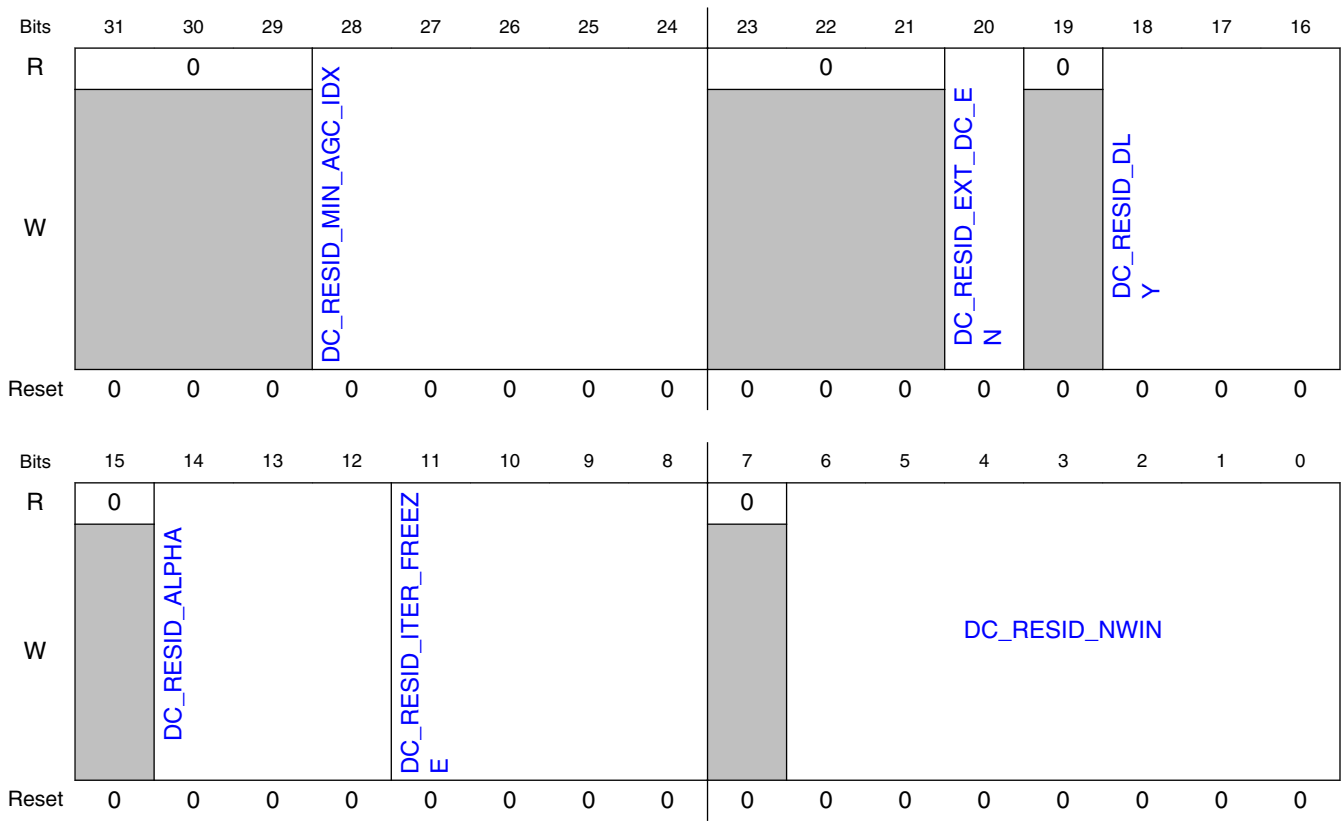
Field	Function
31-26 —	Reserved.
25 AGC_DCOC_START_PT	AGC DCOC Start Point When set, the starting value for the AGC gain table index after DCOC calibration will be AGC_MAN_IDX instead of index 26.
24 AGC_MAN_IDX_EN	AGC Manual Index Enable When set, the AGC gain table index is overridden using AGC_MAN_IDX.
23-21 —	Reserved.
20-16 AGC_MAN_IDX	AGC Manual Index AGC gain table index override value (when AGC_MAN_IDX_EN is set), or AGC gain table index starting point after DCOC calibration (when AGC_DCOC_START_PT is set).
15-0 —	Reserved.

45.3.3.3.1.76 DC Residual Control (DC_RESID_CTRL)

45.3.3.3.1.76.1 Offset

Register	Offset
DC_RESID_CTRL	1D4h

45.3.3.3.1.76.2 Diagram



45.3.3.3.1.76.3 Fields

Field	Function
31-29 —	Reserved.
28-24 DC_RESID_MIN_AGC_IDX	DC Residual Minimum AGC Table Index Specifies the minimum AGC table index value at which DC residual can be enabled. E.g., if this is 5'd0, then the DC residual is enabled for all AGC gain table indexes (assuming RX_DC_RESID_EN is set) if this is 5'd20, then tracking is enabled for AGC gain table indexes 20-26 (assuming RX_DC_RESID_EN is set).
23-21 —	Reserved.

Table continues on the next page...

Field	Function
20 DC_RESID_EXT_DC_EN	DC Residual External DC Enable 0b - External DC disable. The DC Residual activates at a delay specified by DC_RESID_DLY after an AGC gain change pulse. The DC Residual is initialized with a DC offset of 0. 1b - External DC enable. The DC residual activates after the DCOC's tracking hold timer expires. The DC Residual is initialized with the DC estimate from the DCOC tracking estimator.
19 —	Reserved.
18-16 DC_RESID_DLY	DC Residual Delay The delay from activation of the DC residual block to the time when it starts processing. For a 32MHz reference clock, the delay in microseconds matches the value programmed. For other clock frequencies, the delay is scaled based on the clock period. Supported values: 0-7. This delay is only used when EXT_DC_EN=0.
15 —	Reserved.
14-12 DC_RESID_ALPHA	DC Residual Alpha The Alpha parameter controls the rate at which the DC estimate is updated. The update factor is 2^(-Alpha).
11-8 DC_RESID_ITER_FREEZE	DC Residual Iteration Freeze Number of windows of DC_RESID_NWIN samples before the DC is frozen. Supported values: 1-8
7 —	Reserved.
6-0 DC_RESID_NWIN	DC Residual NWIN Number of samples in a window.

45.3.3.3.1.77 DC Residual Estimate (DC_RESID_EST)

45.3.3.3.1.77.1 Offset

Register	Offset
DC_RESID_EST	1D8h

45.3.3.3.1.77.2 Diagram

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

45.3.3.3.1.77.3 Fields

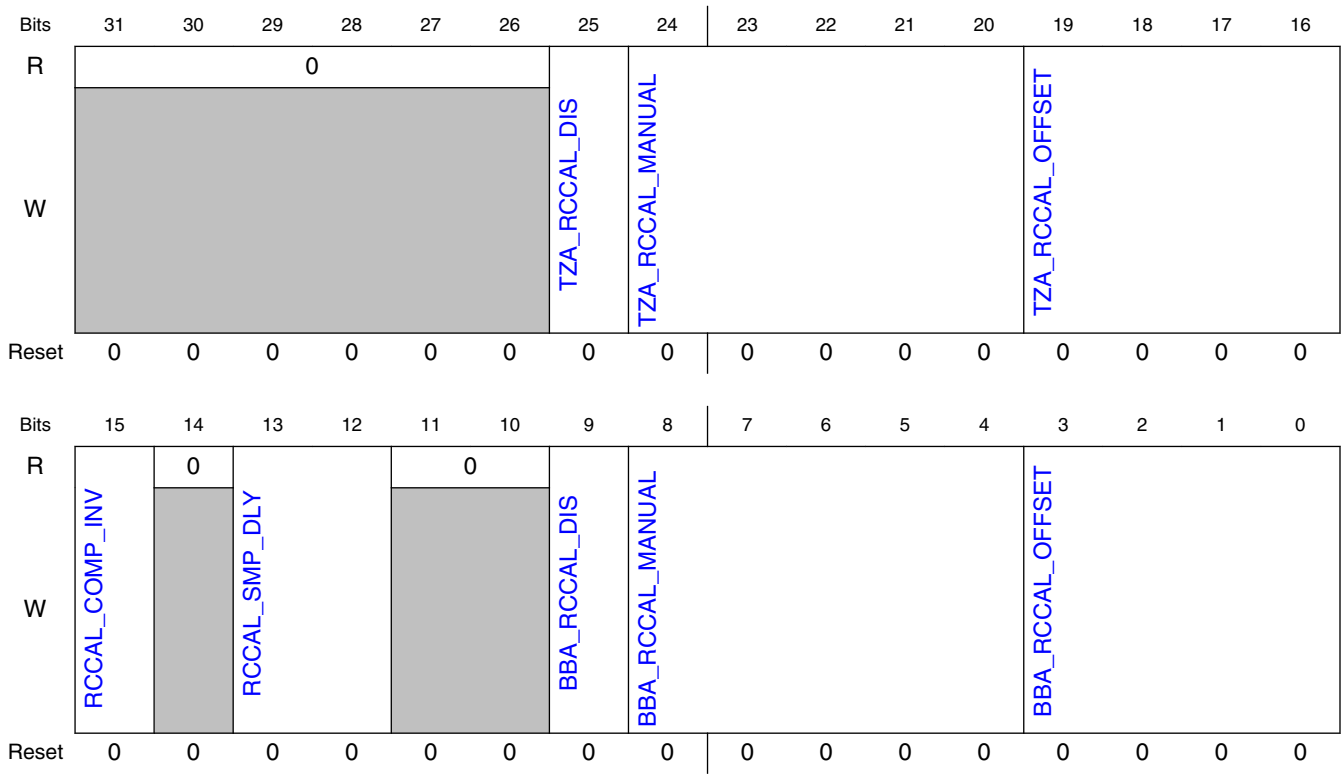
Field	Function
31-29 —	Reserved.
28-16 DC_RESID_OFFSET_Q	DC Residual Offset Q Reflects the current DC residual offset estimate for Q channel. Format is s11.1. This is provided for debug and characterization purposes only.
15-13 —	Reserved.
12-0 DC_RESID_OFFSET_I	DC Residual Offset I Reflects the current DC residual offset estimate for I channel. Format is s11.1. This is provided for debug and characterization purposes only.

45.3.3.3.1.78 RX RC Calibration Control0 (RX_RCCAL_CTRL0)

45.3.3.3.1.78.1 Offset

Register	Offset
RX_RCCAL_CTRL0	1DCh

45.3.3.3.1.78.2 Diagram



45.3.3.3.1.78.3 Fields

Field	Function
31-26 —	Reserved.
25 TZA_RCCAL_DIS	TZA RC Calibration Disable 0b - TZA RC Calibration is enabled 1b - TZA RC Calibration is disabled
24-20 TZA_RCCAL_MANUAL	TZA RC Calibration manual value If TZA_RCCAL_DIS bit is set, this value is used for the TZA calibration.
19-16 TZA_RCCAL_OFFSET	TZA RC Calibration value offset Offset added to the RC Calibration code to determine the calibration value used for the TZA. Format is s3.
15 RCCAL_COMP_INV	RC Calibration comp_out Invert 0b - The comp_out signal polarity is NOT inverted 1b - The comp_out signal polarity is inverted
14 —	Reserved.
13-12	RC Calibration Sample Delay

Table continues on the next page...

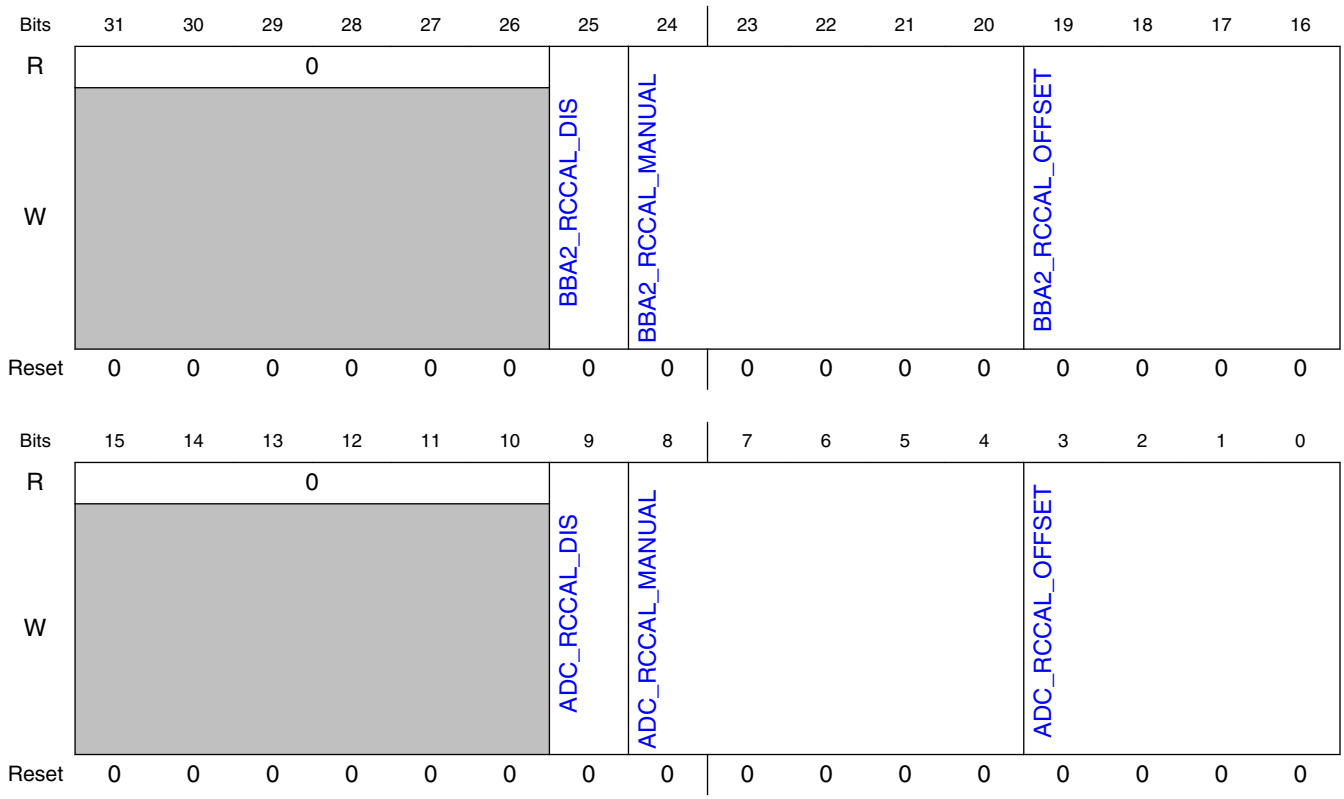
Field	Function
RCCAL_SMP_DLY	00b - The comp_out signal is sampled 0 clk cycle after sample signal is deasserted 01b - The comp_out signal is sampled 1 clk cycle after sample signal is deasserted 10b - The comp_out signal is sampled 2 clk cycle after sample signal is deasserted 11b - The comp_out signal is sampled 3 clk cycle after sample signal is deasserted
11-10 —	Reserved.
9 BBA_RCCAL_DIS	BBA RC Calibration Disable 0b - BBA RC Calibration is enabled 1b - BBA RC Calibration is disabled
8-4 BBA_RCCAL_MANUAL	BBA RC Calibration manual value If BBA_RCCAL_DIS bit is set, this value is used for the BBA calibration.
3-0 BBA_RCCAL_OFFSET	BBA RC Calibration value offset Offset added to the RC Calibration code to determine the calibration value used for the BBA. Format is s3.

45.3.3.3.1.79 RX RC Calibration Control1 (RX_RCCAL_CTRL1)

45.3.3.3.1.79.1 Offset

Register	Offset
RX_RCCAL_CTRL1	1E0h

45.3.3.3.1.79.2 Diagram



45.3.3.3.1.79.3 Fields

Field	Function
31-26 —	Reserved.
25 BBA2_RCCAL_DIS	BBA2 RC Calibration Disable 0b - BBA2 RC Calibration is enabled 1b - BBA2 RC Calibration is disabled
24-20 BBA2_RCCAL_MANUAL	BBA2 RC Calibration manual value If BBA2_RCCAL_DIS bit is set, this value is used for the BBA2 calibration.
19-16 BBA2_RCCAL_OFFSET	BBA2 RC Calibration value offset Offset added to the RC Calibration code to determine the calibration value used for the BBA2. Format is s3.
15-10 —	Reserved.
9 ADC_RCCAL_DIS	ADC RC Calibration Disable 0b - ADC RC Calibration is enabled 1b - ADC RC Calibration is disabled
8-4	ADC RC Calibration manual value

Table continues on the next page...

Field	Function
ADC_RCCAL_MANUAL	If ADC_RCCAL_DIS bit is set, this value is used for the ADC calibration.
3-0 ADC_RCCAL_OFFSET	ADC RC Calibration value offset Offset added to the RC Calibration code to determine the calibration value used for the ADC. Format is s3.

45.3.3.3.1.80 RX RC Calibration Status (RX_RCCAL_STAT)

45.3.3.3.1.80.1 Offset

Register	Offset
RX_RCCAL_STAT	1E4h

45.3.3.3.1.80.2 Diagram

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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BBA2_RCCAL					ADC_RCCAL					RCCAL_CODE				
W																
Reset	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0

45.3.3.3.1.80.3 Fields

Field	Function
31-26 —	Reserved.
25-21 TZA_RCCAL	TZA RC Calibration The RC Calibration value used for TZA
20-16 BBA_RCCAL	BBA RC Calibration The RC Calibration value used for BBA
15 —	Reserved.
14-10	BBA2 RC Calibration

Table continues on the next page...

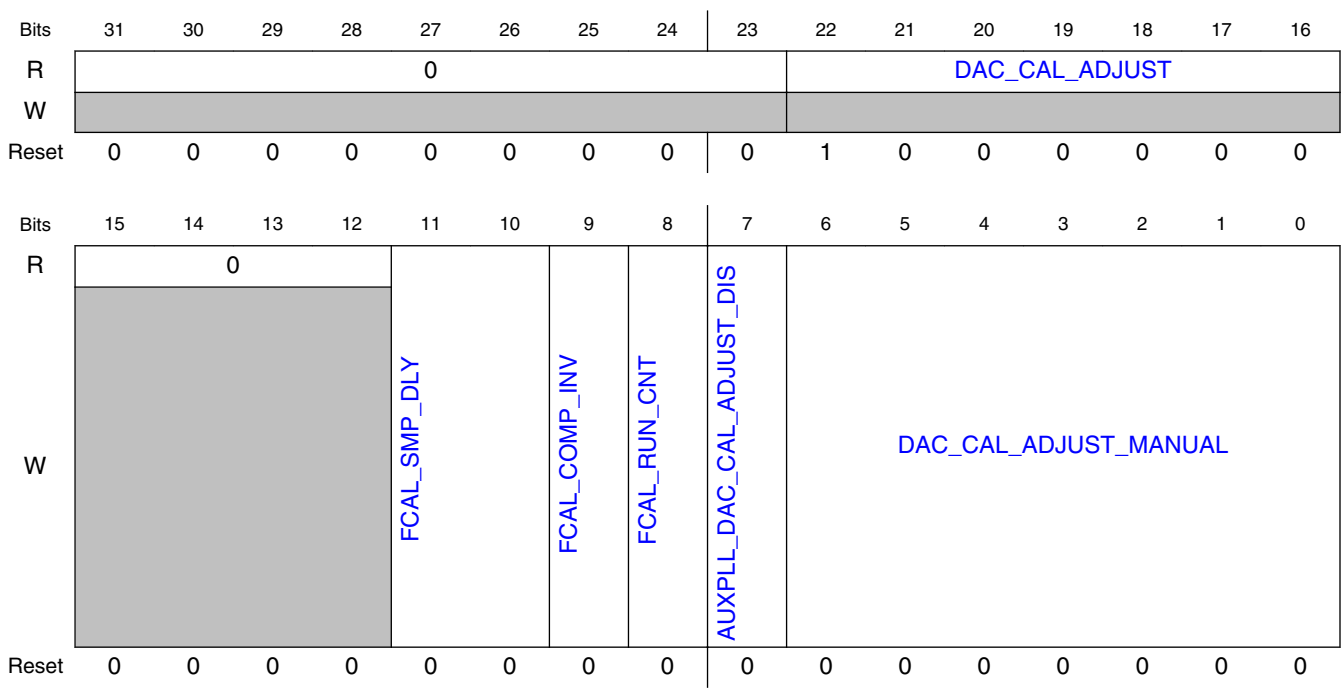
Field	Function
BBA2_RCCAL	The RC Calibration value used for BBA2
9-5 ADC_RCCAL	ADC RC Calibration The RC Calibration value used for ADC
4-0 RCCAL_CODE	RC Calibration code The RC Calibration code currently applied to the calibration circuit

45.3.3.3.1.81 Aux PLL Frequency Calibration Control (AUXPLL_FCAL_CTRL)

45.3.3.3.1.81.1 Offset

Register	Offset
AUXPLL_FCAL_CTRL	1E8h

45.3.3.3.1.81.2 Diagram



45.3.3.3.1.81.3 Fields

Field	Function
31-23	Reserved.

Table continues on the next page...

Field	Function
—	
22-16 DAC_CAL_ADJUST	Aux PLL DAC Calibration Adjust value The DAC calibration adjust value applied to the calibration circuit.
15-12 —	Reserved.
11-10 FCAL_SMP_DLY	Aux PLL Frequency Calibration Sample Delay 00b - The count signal is sampled 1 clk cycle after fcal_run signal is deasserted 01b - The count signal is sampled 2 clk cycle after fcal_run signal is deasserted 10b - The count signal is sampled 3 clk cycle after fcal_run signal is deasserted 11b - The count signal is sampled 4 clk cycle after fcal_run signal is deasserted
9 FCAL_COMP_INV	Aux PLL Frequency Calibration Comparison Invert 0b - (Default) The comparison associated with the count is not inverted. 1b - The comparison associated with the count is inverted
8 FCAL_RUN_CNT	Aux PLL Frequency Calibration Run Count 0b - Run count is 256 clock cycles 1b - Run count is 512 clock cycles
7 AUXPLL_DAC_CAL_ADJUST_DIS	Aux PLL Frequency Calibration Disable 0b - Calibration is enabled 1b - Calibration is disabled
6-0 DAC_CAL_ADJUST_MANUAL	Aux PLL Frequency DAC Calibration Adjust Manual value If AUXPLL_DAC_CAL_ADJUST_DIS bit is set, this DAC calibration adjust value is applied to the calibration circuit.

45.3.3.3.1.82 Aux PLL Frequency Calibration Count 6 (AUXPLL_FCAL_CNT6)

45.3.3.3.1.82.1 Offset

Register	Offset
AUXPLL_FCAL_CNT6	1ECh

45.3.3.3.1.82.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						FCAL_BESTDIFF									
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	0						FCAL_COUNT_6									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.82.3 Fields

Field	Function
31-26 —	Reserved.
25-16 FCAL_BESTDIFF	Aux PLL Frequency Calibration Best Difference The smallest absolute difference between the count value output by the calibration circuit and the expected count value found during the calibration sequence.
15-10 —	Reserved.
9-0 FCAL_COUNT_6	Aux PLL Frequency Calibration Count 6 The count value output by the calibration circuit for calibration phase 6.

45.3.3.3.1.83 Aux PLL Frequency Calibration Count 5 and 4 (AUXPLL_FCAL_CNT5_4)

45.3.3.3.1.83.1 Offset

Register	Offset
AUXPLL_FCAL_CNT5_4	1F0h

45.3.3.3.1.83.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						FCAL_COUNT_5									
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	0						FCAL_COUNT_4									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.83.3 Fields

Field	Function
31-26 —	Reserved.
25-16 FCAL_COUNT_5	Aux PLL Frequency Calibration Count 5 The count value output by the calibration circuit for calibration phase 5.
15-10 —	Reserved.
9-0 FCAL_COUNT_4	Aux PLL Frequency Calibration Count 4 The count value output by the calibration circuit for calibration phase 4.

45.3.3.3.1.84 Aux PLL Frequency Calibration Count 3 and 2 (AUXPLL_FCAL_CNT3_2)

45.3.3.3.1.84.1 Offset

Register	Offset
AUXPLL_FCAL_CNT3_2	1F4h

45.3.3.3.1.84.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						FCAL_COUNT_3									
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	0						FCAL_COUNT_2									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.84.3 Fields

Field	Function
31-26 —	Reserved.
25-16 FCAL_COUNT_3	Aux PLL Frequency Calibration Count 3 The count value output by the calibration circuit for calibration phase 3.
15-10 —	Reserved.
9-0 FCAL_COUNT_2	Aux PLL Frequency Calibration Count 2 The count value output by the calibration circuit for calibration phase 2.

45.3.3.3.1.85 Aux PLL Frequency Calibration Count 1 and 0 (AUXPLL_FCAL_CNT1_0)

45.3.3.3.1.85.1 Offset

Register	Offset
AUXPLL_FCAL_CNT1_0	1F8h

45.3.3.3.1.85.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						FCAL_COUNT_1									
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	0						FCAL_COUNT_0									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.3.3.1.85.3 Fields

Field	Function
31-26 —	Reserved.
25-16 FCAL_COUNT_1	Frequency Calibration Count 1 The count value output by the calibration circuit for calibration phase 1.
15-10 —	Reserved.
9-0 FCAL_COUNT_0	Frequency Calibration Count 0 The count value output by the calibration circuit for calibration phase 0.

45.3.4 Bit Streaming Mode

45.3.4.1 Introduction

The 2.4GHz Radio features a Bit Streaming Mode for BLE. When activated, this feature allows all received BLE packet data, to be serialized and shifted out to external hardware for further processing.

45.3.4.1.1 Overview

Bit Streaming Mode (BSM) allows all received BLE packet data, to be serialized and shifted out to external hardware for further processing. A simple development system can be crafted to consume the BSM outputs and generate packet trace data for all BLE traffic appearing on a network, within range of the receiving device. This enables network-level monitoring and debugging. BSM-for-BLE uses a simple, synchronous 3-wire interface, consisting of BSM_CLK, BSM_DATA, and BSM_FRAME outputs. Packet data is shifted out serially, at the BLE bit rate (1MHz). Signalling is provided on BSM_FRAME to indicate the start of reception. BSM_DATA and BSM_FRAME are synchronous to BSM_CLK. BSM_DATA and BSM_FRAME are valid at rising BSM_CLK, and are intended to be captured on that edge. A single control bit activates or deactivates BSM. Aside from controlling this bit, BSM requires no software support from the host processor. The SoC's BSM outputs are multiplexed with GPIO, so that the pins are available for general purpose use when BSM is disabled. BSM does not interfere with BLE packet processing in any way; it is merely a passive monitoring and debugging tool. BSM, when engaged, will not measurably increase current consumption, since the BSM module operates at a 1MHz clock rate. Only received packets are made available to BSM; packets transmitted by the device do not appear on the BSM interface.

In this BSM implementation, Access Address checking is done in hardware, not software. No protocol engine (link layer) is engaged, so software configures and activates BSM by programming the transceiver (PHY) directly, reducing initialization overhead. And there are no "blind spots" due to link layer SCAN_INTERVAL timeouts. Packets can be received at any time under direct software control.

45.3.4.1.2 Features

- Enables monitoring and debugging of all network traffic within range of the device
- Simple, synchronous 3-wire interface facilitates external hardware development
- Low power consumption
- Non-intrusive design
- Minimal software support required
- Hardware Access Address Correlation
- No Link Layer Overhead

45.3.4.1.3 Block Diagram

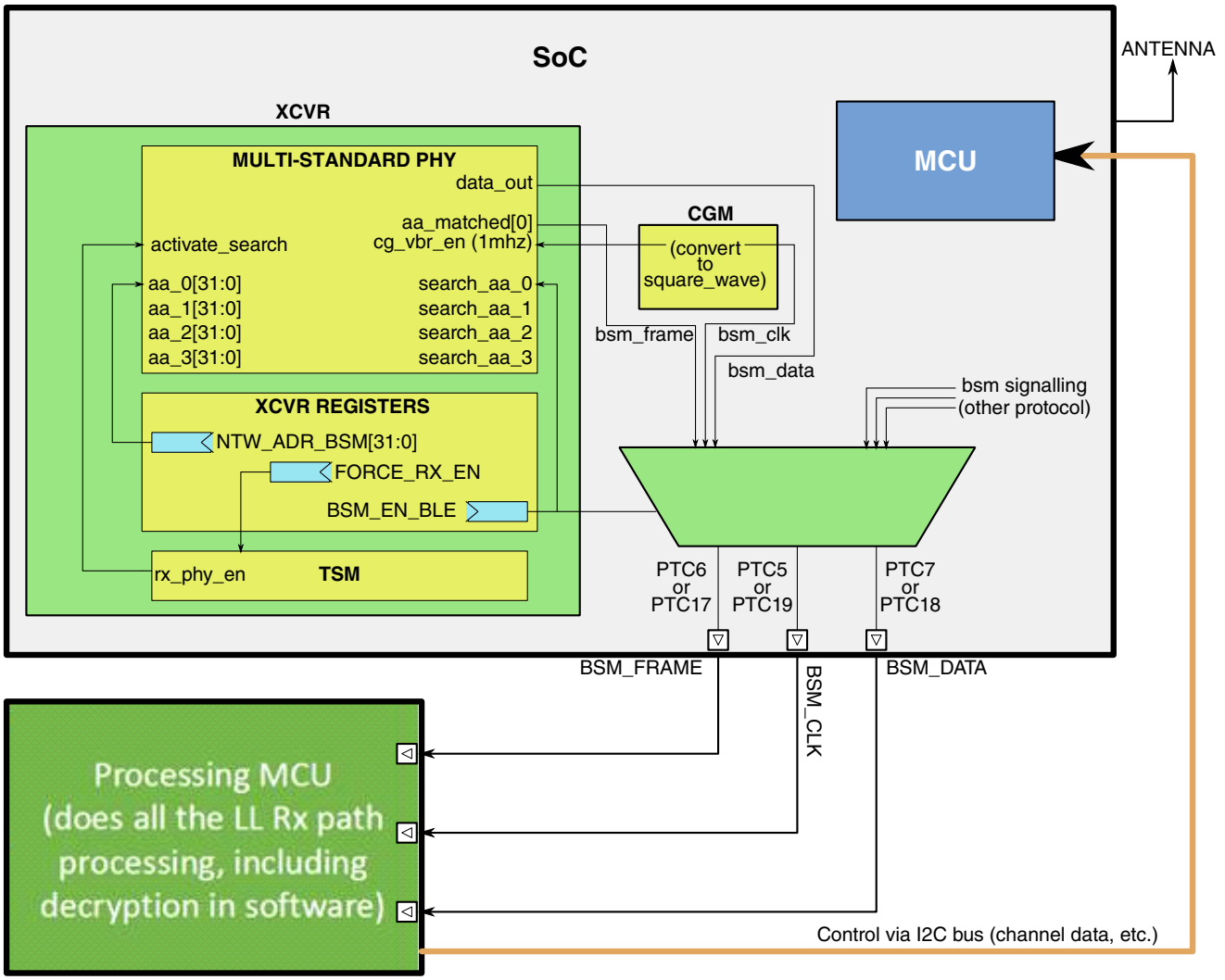


Figure 45-53. BSM Block Diagram

45.3.4.2 External Signal Descriptions

Name	Direction	Size	Description
BSM_DATA	output	1	Serial BLE packet bit stream, LSB-first. Valid on rising edge of BSM_CLK
BSM_FRAME	output	1	Framing signal to indicate the start of reception. Active high
BSM_CLK	output	1	1MHz bit-rate clock. BSM_DATA and BSM_FRAME are

Name	Direction	Size	Descriptiom
			synchronized to BSM_CLK. External device should capture BSM_FRAME and BSM_DATA on rising edge of BSM_CLK

45.3.4.3 Memory Map and Register Definition

BSM-for-BLE is enabled by a single register bit. The bit is BSM_EN_BLE. This bit resides in the PHY_CFG1 register of XCVR address space. Several other programmable configuration parameters pertaining to BSM mode are listed in the table below.

Register	Address	Field	Bit(s)	Desciption
PHY_CFG1	XCVR_BASE + 0x420	BSM_EN_BLE	[5]	1: enable BSM (bit streaming mode) for BLE 0:disable BSM When enabled, the 3 BSM outputs (BSM_DATA, BSM_FRAME, and BSM_BLK) appear on the BSM interface pins of the SoC. At the SoC level, these are alternate, muxed-GPIO pins, so the appropriate port programming is required.
NTW_ADR_BSM	XCVR_BASE + 0x42C	NTW_ADR_BSM[31:0]	[31:0]	PHY will search for this Access Address when BSM_EN_BLE=1
PHY_CFG1	XCVR_BASE + 0x420	BLE_NTW_ADR_THR[2:0]	[30:28]	PHY will tolerate this many bit errors in its search for NTW_ADR_BSM
TSM_CTRL	XCVR_BASE + 0x2C0	FORCE_RX_EN	[3]	Software will set FORCE_RX_EN=1 to launch an RX sequence in search of a packet with an Access Address equal to NTW_ADR_BSM. Software will clear FORCE_RX_EN=0 after the last bit of the packet has been received and processed

Register	Address	Field	Bit(s)	Desciption
				by the application. Clearing this bit ends the RX sequence.

45.3.4.4 Functional Description

The BSM implementation configures the device to enter receive mode and search the airwaves for BLE packets bearing an Access Address matching the one programmed by the application. In this mode, the application communicates directly with the Multi-standard PHY, and no link layers are engaged. Once configured and enabled for searching, BLE packets with an Access Address match will appear at the BSM pins of the device. An Access Address match will result in the device asserting BSM_FRAME=1. At this point, the device will begin shifting out the packet bits serially, LSB first, starting with the packet header (preamble and Access Address will *not* appear on the BSM pins). The BSM pins are a synchronous interface to an external MCU. BSM_FRAME and BSM_DATA bits change on the falling edge of BSM_CLK. The external device must capture BSM_FRAME and BSM_DATA on the rising edge. The application is responsible for all packet processing after the AA match (BSM_FRAME=1), including header parsing and CRC checking. The application terminates the receive operation once the last bit of the packet has been received and processed by the application. The number of bit errors tolerated by the PHY in its correlation of Access Address is programmable, with a default setting of 1 bit error tolerated. The BSM hardware on the device has no knowlege of Bluetooth Low Energy states, so it is up to the application to track BLE state (e.g., Advertising, Initiating), and configure the Access Address accordingly.

The following steps can be used to receive a BLE packet in BSM mode:

1. Configure the RX digital for BLE protocol
2. Configure the PHY for BLE protocol
3. Set PHY_CFG1[BSM_EN_BLE]=1 to enable BSM feature
4. Program the desired Access Address into the NTW_ADR_BSM register
5. Program the desired maximum number of tolerated bit errors into PHY_CFG1[BTLE_NTW_ADR_THR]
6. Set TSM_CTRL[FORCE_RX_EN]=1 to initiate the RX sequence
7. Monitor the BSM interface for BSM_FRAME=1 (Access Address match)
8. After BSM_FRAME=1, use each rising edge of BSM_CLK to capture the next bit of BSM_DATA
9. Shift in and parse the incoming packet on the fly, using the header to determine packet length

10. After the last bit of CRC, terminate the RX sequence by clearing TSM_CTRL[FORCE_RX_EN]=0.

The following diagram depicts the occurrence of a BLE packet which matches the programmed Access Address at the BSM interface:

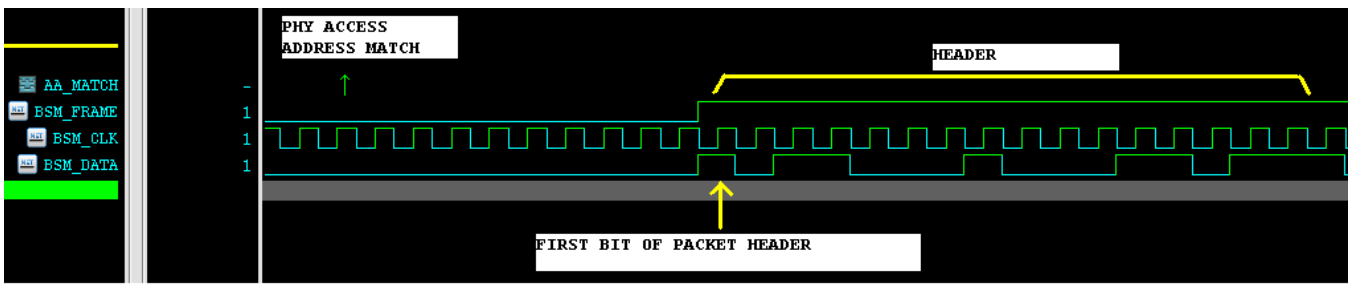


Figure 45-54. BSM Timing Diagram

The following sequence of events is depicted in the diagram

1. AA match occurs internally in the PHY
2. A delay of several microseconds after internal PHY AA match (transparent to application)
3. BSM_FRAME is asserted
4. At the next rising edge of BSM_CLK, first bit of packet header (LSB) is valid
5. Subsequent header bits are shifted out LSB-first on BSM_DATA, valid at each rising edge of BSM_CLK
6. Payload and CRC bits follow the header

The PHY_STATUS register (XCVR_BASE + 0x430) contains several fields which may assist software monitoring the BSM interface:

Field	Bit(s)	Desciption
PREAMBLE_FOUND	[0]	PHY has detected a BLE preamble. This should not be relied on by the application but it may be useful for debug. In the hardware, preamble detection is a prerequisite for Access Address correlation.
AA_SFD_MATCHED	[1]	PHY has detected an Access Address match. After a delay of several microseconds, BSM_FRAME will be asserted and packet data bit shifting will commence.
AA_MATCHED[0]	[4]	Indicates that the Access Address match occurred on the PHY's Network Address 0 input. This will always be the case for BLE.

Software could elect to poll the AA_SFD_MATCHED bit, to determine when to start monitoring the BSM_FRAME, since this bit will transition high several microseconds before BSM_FRAME=1. A procedure for polling AA_SFD_MATCHED as a prerequisite to the monitoring of the BSM interface pins, is as follows:

1. Ascertain the length of the RX Warmup by reading the third byte of the END_OF_SEQ register in XCVR space
2. Set variable **end_of_rx_wu** to this value: **end_of_seq** = (r32(END_OF_SEQ) & 0xFF0000) >> 16
3. Set TSM_CTRL[FORCE_RX_EN]=1 to initiate the RX sequence. (AA_SFD_MATCHED may still be asserted here from the previous packet reception)
4. Poll XCVR_STATUS register in XCVR space to determine when RX Warmup is complete. Set variable **tsm_count** to the LS byte of XCVR_STATUS
5. while (**tsm_count** < **end_of_rx_wu**) {**tsm_count** = (r32(XCVR_STATUS) & 0xFF);}
6. When the while() loop exits, the RX Warmup is complete. BSM_FRAME will assert soon
7. Begin monitoring the BSM pins for incoming BLE packet

45.3.5 Transceiver Sequence Manager

45.3.5.1 Introduction

The TSM is a fully-programmable, multi-protocol transceiver sequence manager, which supports Bluetooth Low Energy, and other 2.4GHz transceivers.

45.3.5.1.1 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 67 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.

45.3.5.1.2 Features

The TSM includes the following features:

- TX warmup and warmdown sequence and RX warmup and warmdown sequence sequence for every controllable output.
- Programmable sequence length from 1 to 254 microseconds for each sequence in 1 microsecond increments.
- 67 total outputs.
- Multi-protocol
- PA Target Power Selection
- PA Ramping
- Ability to place TSM outputs under software control for test and non-mission mode validation.
- Ability to abort sequences upon PLL unlock condition.
- Fast Warmup Capability for TX and RX
- Programmable breakpoint capability to halt any sequence temporarily.

45.3.5.1.3 Block diagram

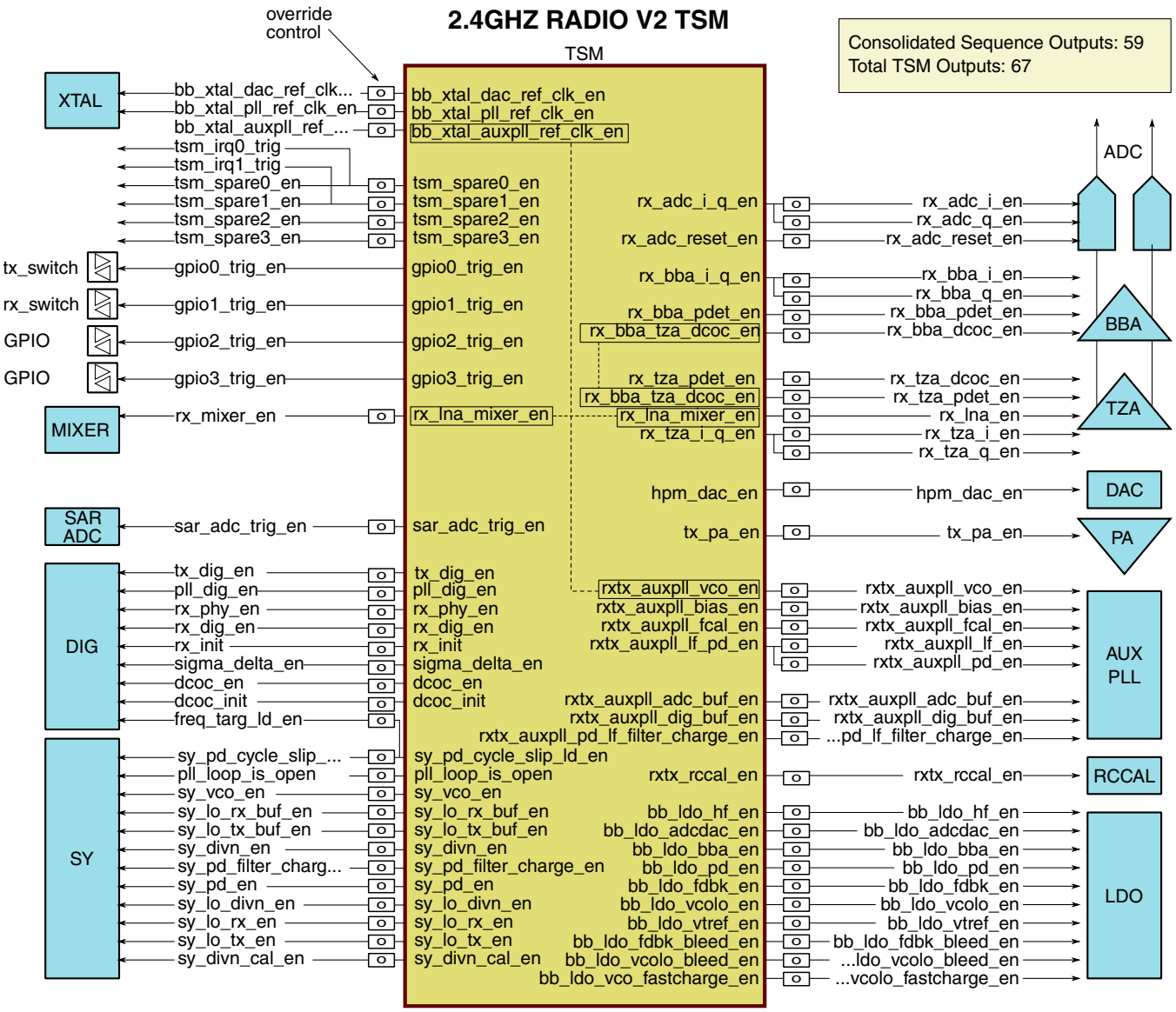


Figure 45-55. Block diagram

45.3.5.2 Functional description

45.3.5.2.1 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). 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). 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). 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.

45.3.5.2.2 TSM-Controlled Outputs

The TSM has 67 timing-controlled outputs, for enabling and control of the various transceiver blocks. The outputs are shown in the following diagram.

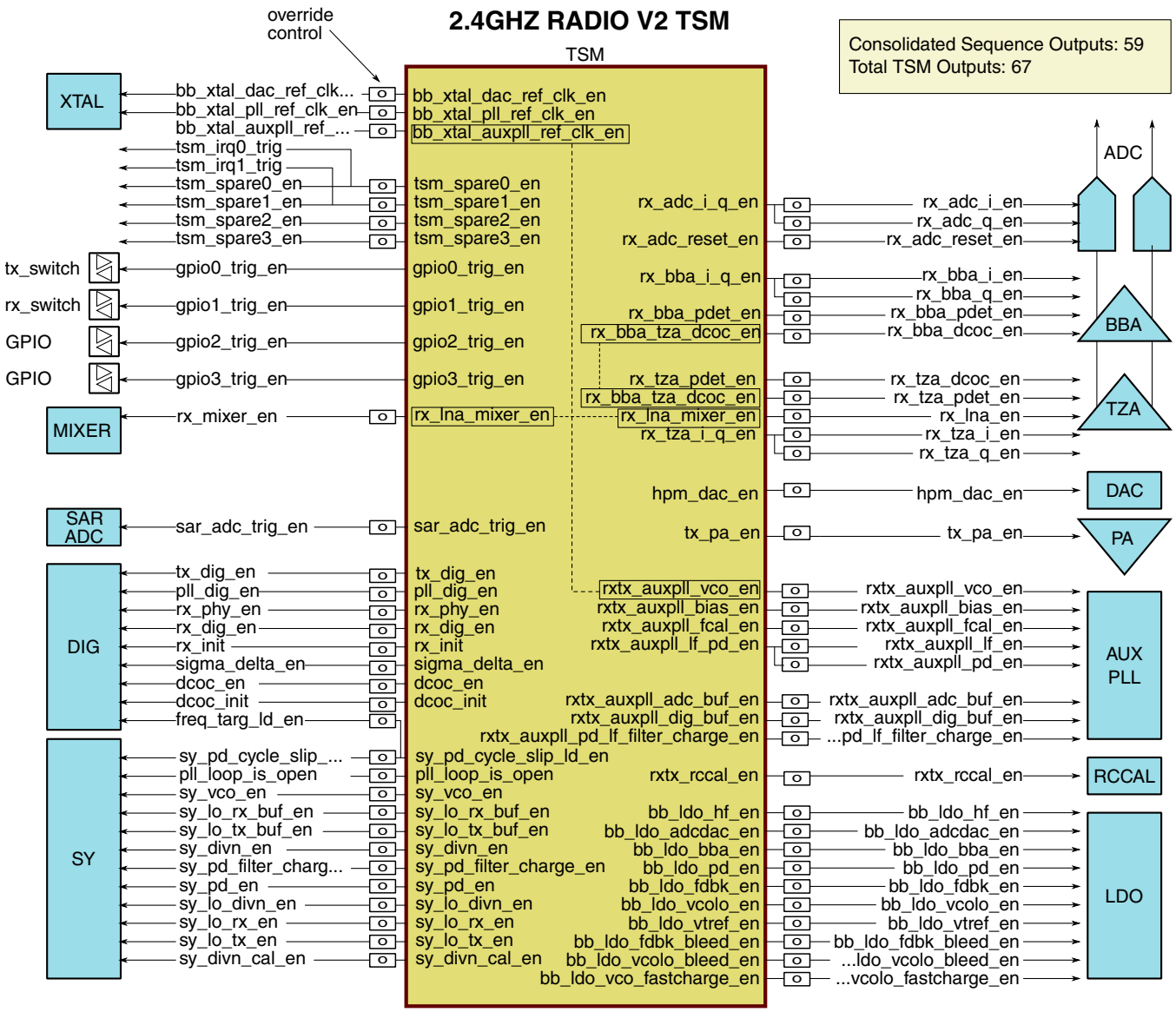


Figure 45-56. 2.4GHZ RADIO TSM

In some cases, some of the 67 outputs are grouped together, for outputs which need identical timing. For example, rx_bba_i_en and rx_bba_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 67 TSM-controlled outputs at any time. As a consequence of the groupings, there are only 59 sets of timing registers, to control the 67 outputs. The table below, lists the 67 TSM-controlled outputs, and for the grouped outputs, indicates the signal which controls the timing for the entire group.

Signal Index	67 TSM-CONTROLLED OUTPUTS	SIGNAL CONTROLLING GROUP TIMING
0	bb_ldo_hf_en	bb_ldo_hf_en
1	bb_ldo_adcdac_en	bb_ldo_adcdac_en
2	bb_ldo_bba_en	bb_ldo_bba_en
3	bb_ldo_pd_en	bb_ldo_pd_en
4	bb_ldo_fdbk_en	bb_ldo_fdbk_en
5	bb_ldo_vcolo_en	bb_ldo_vcolo_en
6	bb_ldo_vtref_en	bb_ldo_vtref_en
7	bb_ldo_fdbk_bleed_en	bb_ldo_fdbk_bleed_en
8	bb_ldo_vcolo_bleed_en	bb_ldo_vcolo_bleed_en
9	bb_ldo_vcolo_fastcharge_en	bb_ldo_vcolo_fastcharge_en
10	bb_xtal_pll_ref_clk_en	bb_xtal_pll_ref_clk_en
11	bb_xtal_dac_ref_clk_en	bb_xtal_dac_ref_clk_en
12	bb_xtal_auxpll_ref_clk_en	rxtx_auxpll_vco_ref_clk_en
13	pll_loop_is_open	pll_loop_is_open
14	sy_pd_cycle_slip_ld_en	sy_pd_cycle_slip_ld_ft_en
15	sy_vco_en	sy_vco_en
16	sy_lo_rx_buf_en	sy_lo_rx_buf_en
17	sy_lo_tx_buf_en	sy_lo_tx_buf_en
18	sy_divn_en	sy_divn_en
19	sy_pd_filter_charge_en	sy_pd_filter_charge_en
20	sy_pd_en	sy_pd_en
21	sy_lo_divn_en	sy_lo_divn_en
22	sy_lo_rx_en	sy_lo_rx_en
23	sy_lo_tx_en	sy_lo_tx_en
24	sy_divn_cal_en	sy_divn_cal_en
25	rx_mixer_en	rx_lna_mixer_en
26	tx_pa_en	tx_pa_en
27	rx_adc_i_en	rx_adc_i_q_en
28	rx_adc_q_en	rx_adc_i_q_en
29	rx_adc_reset_en	rx_adc_reset_en
30	rx_bba_i_en	rx_bba_i_q_en
31	rx_bba_q_en	rx_bba_i_q_en
32	rx_bba_pdet_en	rx_bba_pdet_en
33	rx_bba_dcoc_en	rx_bba_tza_dcoc_en
34	rx_lna_en	rx_lna_mixer_en
35	rx_tza_i_en	rx_tza_i_q_en
36	rx_tza_q_en	rx_tza_i_q_en
37	rx_tza_pdet_en	rx_tza_pdet_en
38	rx_tza_dcoc_en	rx_bba_tza_dcoc_en
39	pll_dig_en	pll_dig_en

Table continues on the next page...

Signal Index	67 TSM-CONTROLLED OUTPUTS	SIGNAL CONTROLLING GROUP TIMING
40	tx_dig_en	tx_dig_en
41	rx_dig_en	rx_dig_en
42	rx_init	rx_init
43	sigma_delta_en	sigma_delta_en
44	rx_phy_en	rx_phy_en
45	dcoc_en	dcoc_en
46	dcoc_init	dcoc_init
47	freq_targ_ld_en	sy_pd_cycle_slip_ld_ft_en
48	sar_adc_trig_en	sar_adc_trig_en
49	tsm_spare0_en	tsm_spare0_en
50	tsm_spare1_en	tsm_spare1_en
51	tsm_spare2_en	tsm_spare2_en
52	tsm_spare3_en	tsm_spare3_en
53	gpio0_trig_en	gpio0_trig_en
54	gpio1_trig_en	gpio1_trig_en
55	gpio2_trig_en	gpio2_trig_en
56	gpio3_trig_en	gpio3_trig_en
57	rxtx_auxpll_bias_en	rxtx_auxpll_bias_en
58	rxtx_auxpll_vco_en	rxtx_auxpll_vco_ref_clk_en
59	rxtx_auxpll_fcal_en	rxtx_auxpll_fcal_en
60	rxtx_auxpll_lf_en	rxtx_auxpll_lf_pd_en
61	rxtx_auxpll_pd_en	rxtx_auxpll_lf_pd_en
62	rxtx_auxpll_pd_lf_filter_charge_en	rxtx_auxpll_pd_lf_filter_charge_en
63	rxtx_auxpll_adc_buf_en	rxtx_auxpll_adc_buf_en
64	rxtx_auxpll_dig_buf_en	rxtx_auxpll_dig_buf_en
65	rxtx_rccal_en	rxtx_rccal_en
66	tx_hpm_dac_en	tx_hpm_dac_en

For example, TSM-controlled ouputs rx_bba_i_en, and rx_bba_q_en, are both members of the same group. The timing registers to program the timing on these outputs (RX_BBA_I_Q_TX_HI, RX_BBA_I_Q_TX_LO, RX_BBA_I_Q_RX_HI, RX_BBA_I_Q_RX_LO), control the timing for both 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 67 outputs is shown.

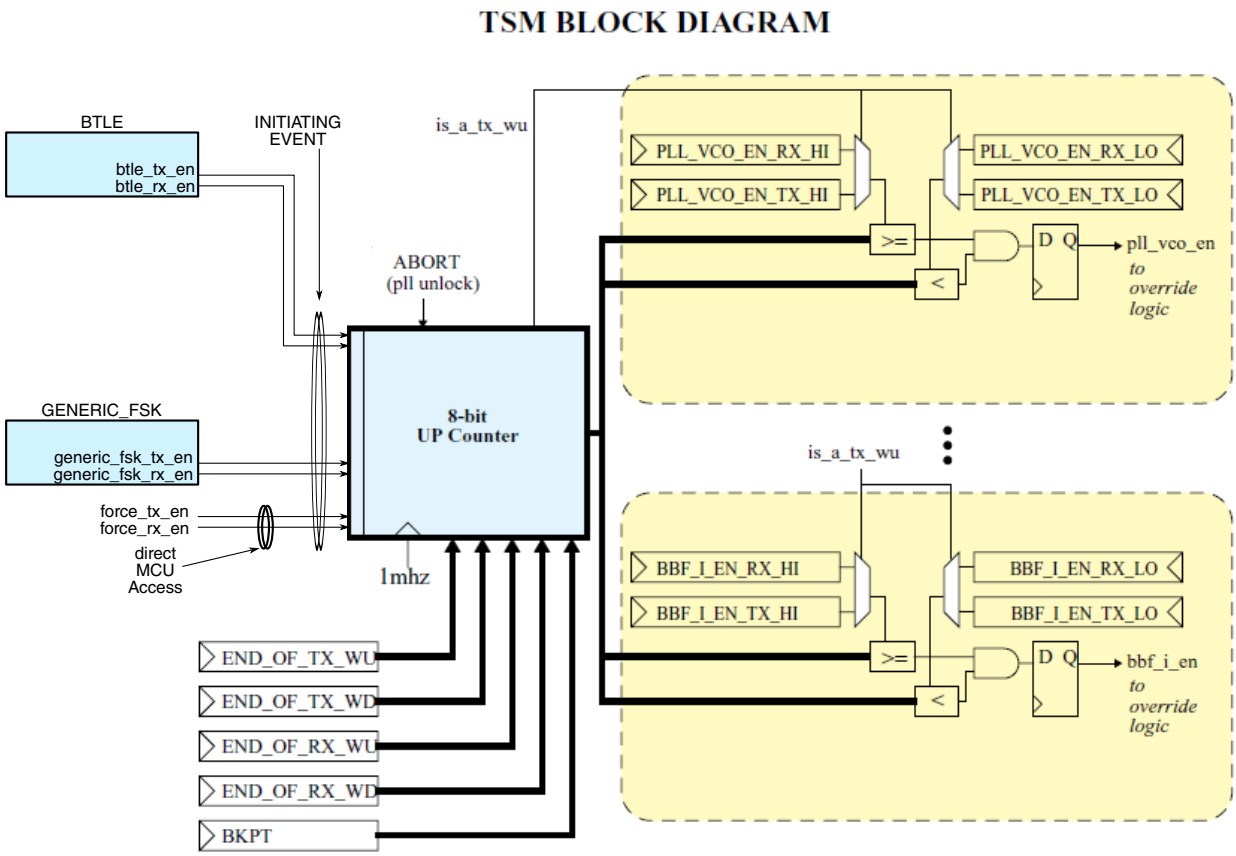


Figure 45-57. 2.4GHZ RADIO TSM BLOCK DIAGRAM

45.3.5.2.3 Timing Registers

There are 59 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 59 sets, the following table lists the register name (TSM_TIMING00 – TSM_TIMING58), and 4 timing register names associated with each.

REGISTER NAME	BITS [31:24]	BITS [23:16]	BITS [15:8]	BITS [7:0]
TSM_TIMING00	BB_LDO_HF_EN_RX_LO[7:0]	BB_LDO_HF_EN_RX_HI[7:0]	BB_LDO_HF_EN_TX_LO[7:0]	BB_LDO_HF_EN_TX_HI[7:0]
TSM_TIMING01	BB_LDO_ADCDAC_EN_RX_LO[7:0]	BB_LDO_ADCDAC_EN_RX_HI[7:0]	BB_LDO_ADCDAC_EN_TX_LO[7:0]	BB_LDO_ADCDAC_EN_TX_HI[7:0]

Table continues on the next page...

REGISTER NAME	BITS [31:24]	BITS [23:16]	BITS [15:8]	BITS [7:0]
TSM_TIMING02	BB_LDO_BBA_EN_RX_LO[7:0]	BB_LDO_BBA_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING03	BB_LDO_PD_EN_RX_LO[7:0]	BB_LDO_PD_EN_RX_HI[7:0]	BB_LDO_PD_EN_TX_LO[7:0]	BB_LDO_PD_EN_TX_HI[7:0]
TSM_TIMING04	BB_LDO_FDBK_EN_RX_LO[7:0]	BB_LDO_FDBK_EN_RX_HI[7:0]	BB_LDO_FDBK_EN_TX_LO[7:0]	BB_LDO_FDBK_EN_TX_HI[7:0]
TSM_TIMING05	BB_LDO_VCOLO_EN_RX_LO[7:0]	BB_LDO_VCOLO_EN_RX_HI[7:0]	BB_LDO_VCOLO_EN_TX_LO[7:0]	BB_LDO_VCOLO_EN_TX_HI[7:0]
TSM_TIMING06	BB_LDO_VTREF_EN_RX_LO[7:0]	BB_LDO_VTREF_EN_RX_HI[7:0]	BB_LDO_VTREF_EN_TX_LO[7:0]	BB_LDO_VTREF_EN_TX_HI[7:0]
TSM_TIMING07	BB_LDO_FDBK_BLEED_EN_RX_LO[7:0]	BB_LDO_FDBK_BLEED_EN_RX_HI[7:0]	BB_LDO_FDBK_BLEED_EN_TX_LO[7:0]	BB_LDO_FDBK_BLEED_EN_TX_HI[7:0]
TSM_TIMING08	BB_LDO_VCOLO_BLEED_EN_RX_LO[7:0]	BB_LDO_VCOLO_BLEED_EN_RX_HI[7:0]	BB_LDO_VCOLO_BLEED_EN_TX_LO[7:0]	BB_LDO_VCOLO_BLEED_EN_TX_HI[7:0]
TSM_TIMING09	BB_LDO_VCOLO_FASTCHARGE_EN_RX_LO[7:0]	BB_LDO_VCOLO_FASTCHARGE_EN_RX_HI[7:0]	BB_LDO_VCOLO_FASTCHARGE_EN_TX_LO[7:0]	BB_LDO_VCOLO_FASTCHARGE_EN_TX_HI[7:0]
TSM_TIMING10	BB_XTAL_PLL_REF_CLK_EN_RX_LO[7:0]	BB_XTAL_PLL_REF_CLK_EN_RX_HI[7:0]	BB_XTAL_PLL_REF_CLK_EN_TX_LO[7:0]	BB_XTAL_PLL_REF_CLK_EN_TX_HI[7:0]
TSM_TIMING11	n/a	n/a	BB_XTAL_DAC_REF_CLK_EN_TX_LO[7:0]	BB_XTAL_DAC_REF_CLK_EN_TX_HI[7:0]
TSM_TIMING12	RXTX_AUXPLL_VCO_REF_CLK_EN_RX_LO[7:0]	RXTX_AUXPLL_VCO_REF_CLK_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING13	PLL_LOOP_IS_OPEN_RX_LO[7:0]	PLL_LOOP_IS_OPEN_RX_HI[7:0]	PLL_LOOP_IS_OPEN_TX_LO[7:0]	PLL_LOOP_IS_OPEN_TX_HI[7:0]
TSM_TIMING14	SY_PD_CYCLE_SLIP_LD_FT_EN_RX_LO[7:0]	SY_PD_CYCLE_SLIP_LD_FT_EN_RX_HI[7:0]	SY_PD_CYCLE_SLIP_LD_FT_EN_TX_LO[7:0]	SY_PD_CYCLE_SLIP_LD_FT_EN_TX_HI[7:0]
TSM_TIMING15	SY_VCO_EN_RX_LO[7:0]	SY_VCO_EN_RX_HI[7:0]	SY_VCO_EN_TX_LO[7:0]	SY_VCO_EN_TX_HI[7:0]
TSM_TIMING16	SY_LO_RX_BUF_EN_RX_LO[7:0]	SY_LO_RX_BUF_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING17	n/a	n/a	SY_LO_TX_BUF_EN_TX_LO[7:0]	SY_LO_TX_BUF_EN_TX_HI[7:0]
TSM_TIMING18	SY_DIVN_EN_RX_LO[7:0]	SY_DIVN_EN_RX_HI[7:0]	SY_DIVN_EN_TX_LO[7:0]	SY_DIVN_EN_TX_HI[7:0]
TSM_TIMING19	SY_PD_FILTER_CHARGE_EN_RX_LO[7:0]	SY_PD_FILTER_CHARGE_EN_RX_HI[7:0]	SY_PD_FILTER_CHARGE_EN_TX_LO[7:0]	SY_PD_FILTER_CHARGE_EN_TX_HI[7:0]
TSM_TIMING20	SY_PD_EN_RX_LO[7:0]	SY_PD_EN_RX_HI[7:0]	SY_PD_EN_TX_LO[7:0]	SY_PD_EN_TX_HI[7:0]
TSM_TIMING21	SY_LO_DIVN_EN_RX_LO[7:0]	SY_LO_DIVN_EN_RX_HI[7:0]	SY_LO_DIVN_EN_TX_LO[7:0]	SY_LO_DIVN_EN_TX_HI[7:0]
TSM_TIMING22	SY_LO_RX_EN_RX_LO[7:0]	SY_LO_RX_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING23	n/a	n/a	SY_LO_TX_EN_TX_LO[7:0]	SY_LO_TX_EN_TX_HI[7:0]

Table continues on the next page...

REGISTER NAME	BITS [31:24]	BITS [23:16]	BITS [15:8]	BITS [7:0]
TSM_TIMING24	SY_DIVN_CAL_EN_RX_LO[7:0]	SY_DIVN_CAL_EN_RX_HI[7:0]	SY_DIVN_CAL_EN_TX_LO[7:0]	SY_DIVN_CAL_EN_TX_HI[7:0]
TSM_TIMING25	RX_LNA_MIXER_EN_RX_LO[7:0]	RX_LNA_MIXER_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING26	n/a	n/a	TX_PA_EN_TX_LO[7:0]	TX_PA_EN_TX_HI[7:0]
TSM_TIMING27	RX_ADC_I_Q_EN_RX_LO[7:0]	RX_ADC_I_Q_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING28	RX_ADC_RESET_EN_RX_LO[7:0]	RX_ADC_RESET_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING29	RX_BBA_I_Q_EN_RX_LO[7:0]	RX_BBA_I_Q_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING30	RX_BBA_PDET_EN_RX_LO[7:0]	RX_BBA_PDET_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING31	RX_BBA_TZA_DCOC_EN_RX_LO[7:0]	RX_BBA_TZA_DCOC_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING32	RX_TZA_I_Q_EN_RX_LO[7:0]	RX_TZA_I_Q_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING33	RX_TZA_PDET_EN_RX_LO[7:0]	RX_TZA_PDET_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING34	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_TIMING35	n/a	n/a	TX_DIG_EN_TX_LO[7:0]	TX_DIG_EN_TX_HI[7:0]
TSM_TIMING36	RX_DIG_EN_RX_LO[7:0]	RX_DIG_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING37	RX_INIT_RX_LO[7:0]	RX_INIT_RX_HI[7:0]	n/a	n/a
TSM_TIMING38	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_TIMING39	RX_PHY_EN_RX_LO[7:0]	RX_PHY_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING40	DCOC_EN_RX_LO[7:0]	DCOC_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING41	DCOC_INIT_RX_LO[7:0]	DCOC_INIT_RX_HI[7:0]	n/a	n/a
TSM_TIMING42	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_TIMING43	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_TIMING44	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_TIMING45	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_TIMING46	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_TIMING47	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]

Table continues on the next page...

REGISTER NAME	BITS [31:24]	BITS [23:16]	BITS [15:8]	BITS [7:0]
TSM_TIMING48	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_TIMING49	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_TIMING50	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]
TSM_TIMING51	RXTX_AUXPLL_BIAS_EN_RX_LO[7:0]	RXTX_AUXPLL_BIAS_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING52	RXTX_AUXPLL_FCAL_EN_RX_LO[7:0]	RXTX_AUXPLL_FCAL_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING53	RXTX_AUXPLL_LF_PD_EN_RX_LO[7:0]	RXTX_AUXPLL_LF_PD_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING54	RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_RX_LO[7:0]	RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING55	RXTX_AUXPLL_ADC_BUF_EN_RX_LO[7:0]	RXTX_AUXPLL_ADC_BUF_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING56	RXTX_AUXPLL_DIG_BUF_EN_RX_LO[7:0]	RXTX_AUXPLL_DIG_BUF_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING57	RXTX_RCCAL_EN_RX_LO[7:0]	RXTX_RCCAL_EN_RX_HI[7:0]	n/a	n/a
TSM_TIMING58	n/a	n/a	TX_HPM_DAC_EN_TX_LO[7:0]	TX_HPM_DAC_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.

45.3.5.2.4 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
Generic FSK	generic_tx_en, generic_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).

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.

45.3.5.2.5 Sequence Termination

A sequence may be terminated during the WARMUP or 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.

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.

45.3.5.2.6 Overrides

Each of the 67 TSM-controlled outputs has independent override control, except for the 4 GPIO triggers, and the SAR_ADC_TRIG output. In addition, two special TSM outputs, tx_mode and rx_mode, also have their own override controls, for a total of 67 – 5 + 2 = 64 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 64 override bit-pairs, reside in the TSM_OVRD0, TSM_OVRD1, TSM_OVRD2, and TSM_OVRD3 registers.

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.

45.3.5.2.7 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.

45.3.5.2.8 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

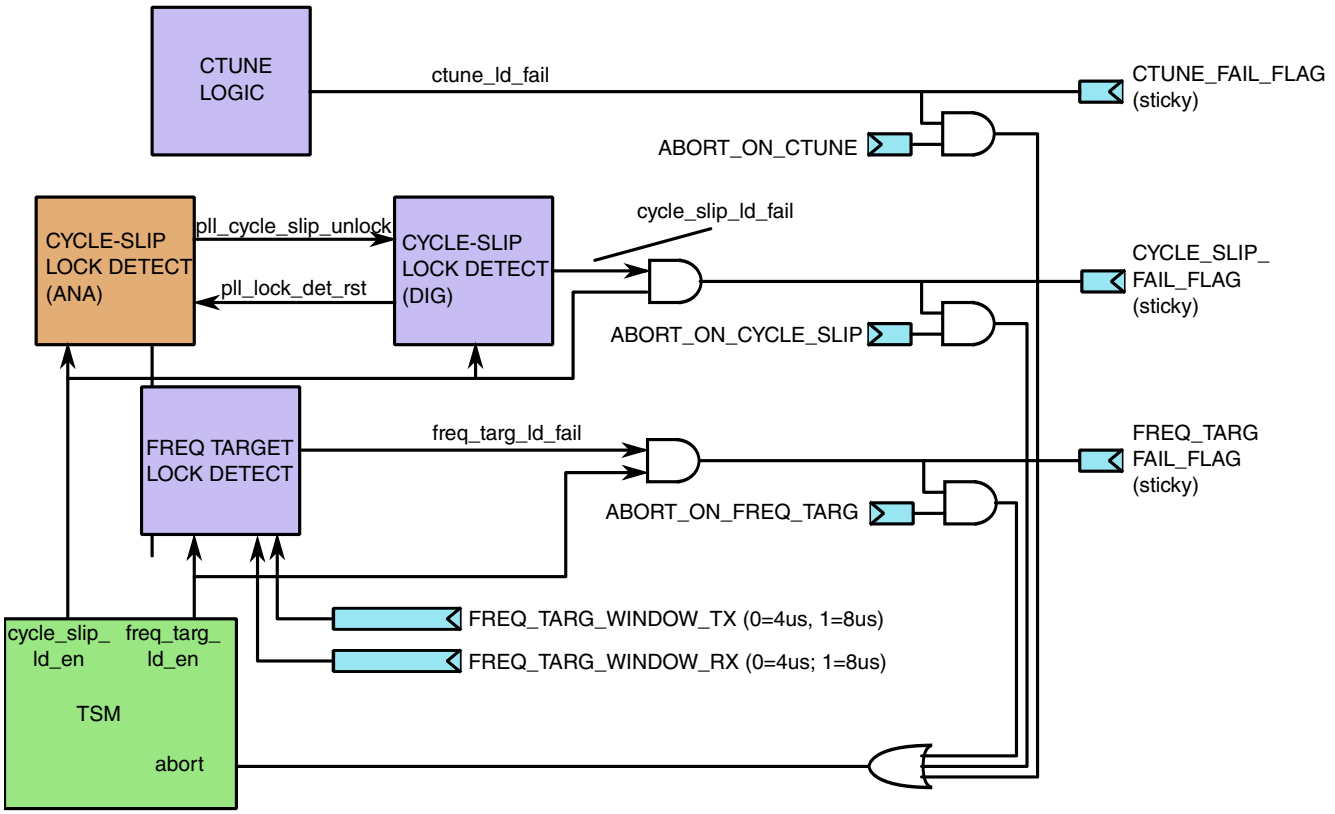


Figure 45-58. PLL UNLOCK MONITORING AND ABORT LOGIC

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_TIMING34	7us	7us
Freq. Target	freq_targ_ld_en	TSM_TIMING14	99us	49us
Cycle Slip	pll_cycle_slip_ld_en	TSM_TIMING14	99us	49us

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.

45.3.5.2.9 Special Handling for GENERIC_FSK

The GENERIC_FSK Link Layer Controller already has built-in handling for PLL unlock conditions, and an interrupt status bit to notify on such an event. The combined-abort signal is blocked from reaching the TSM, and instead is routed to the GENERIC_FSK Command Decoder. The Command Decoder state machine handles the unlock event by deasserting the TX or RX command signal to the TSM (e.g., generic_fsk_tx_en), causing a TSM warmdown to occur. The Generic controller will also set the PLL_UNLOCK_IRQ interrupt status bit in its address space. The diagram below shows the GENERIC_FSK Link Layer handling of the PLL unlock event.

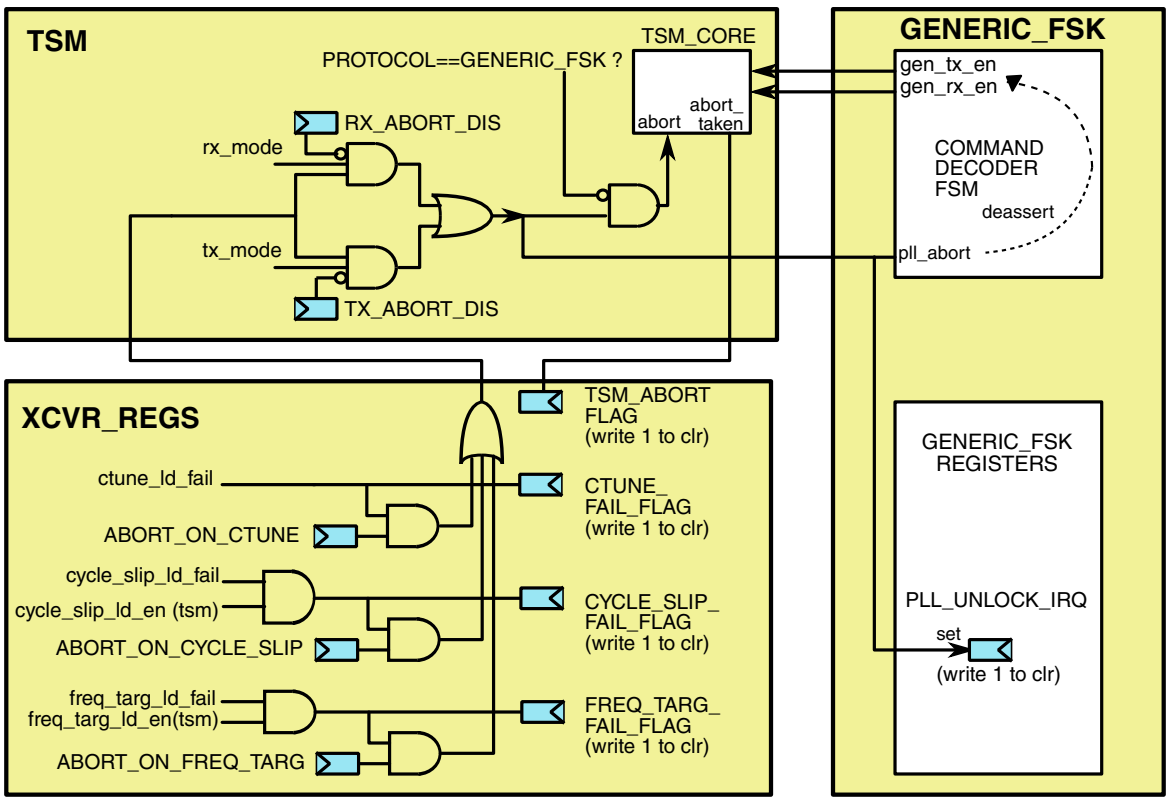


Figure 45-59. GENERIC_FSK PLL UNLOCK HANDLING

For GENERIC_FSK, the TX_ABORT_DIS and RX_ABORT_DIS control bits are still available, and selectively allow/disallow aborting during the TX sequences (TX_ABORT_DIS), or the RX sequences (RX_ABORT_DIS).

45.3.5.2.10 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:

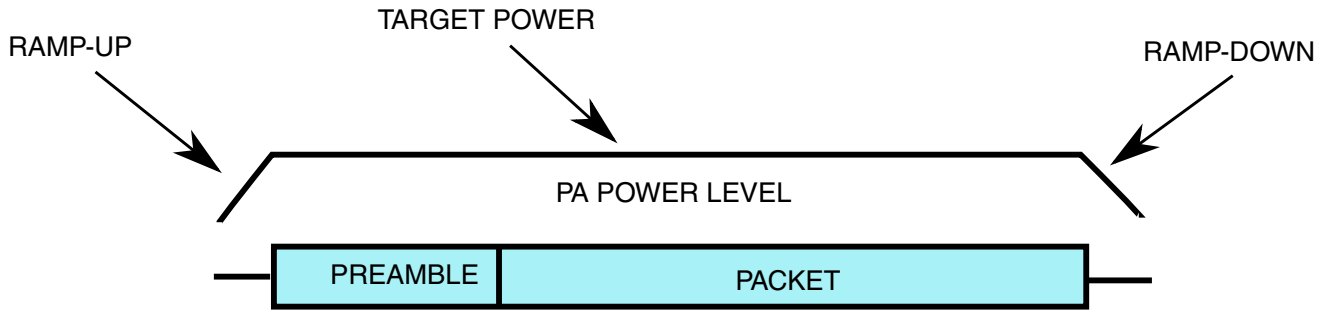


Figure 45-60. PA Target Power

Target power, the steady-state power level that is used to transmit the packet, can be selected from one of the following sources:

1. PA_POWER[5:0] register in XCVR address space
2. BTLE Link Layer (2 distinct power levels supported, for Advertising- and Data-channel Packets)
3. GENERIC_FSK Link Layer (TX_POWER[5:0] register in GENERIC_FSK address space)

The following diagram depicts target power selection.

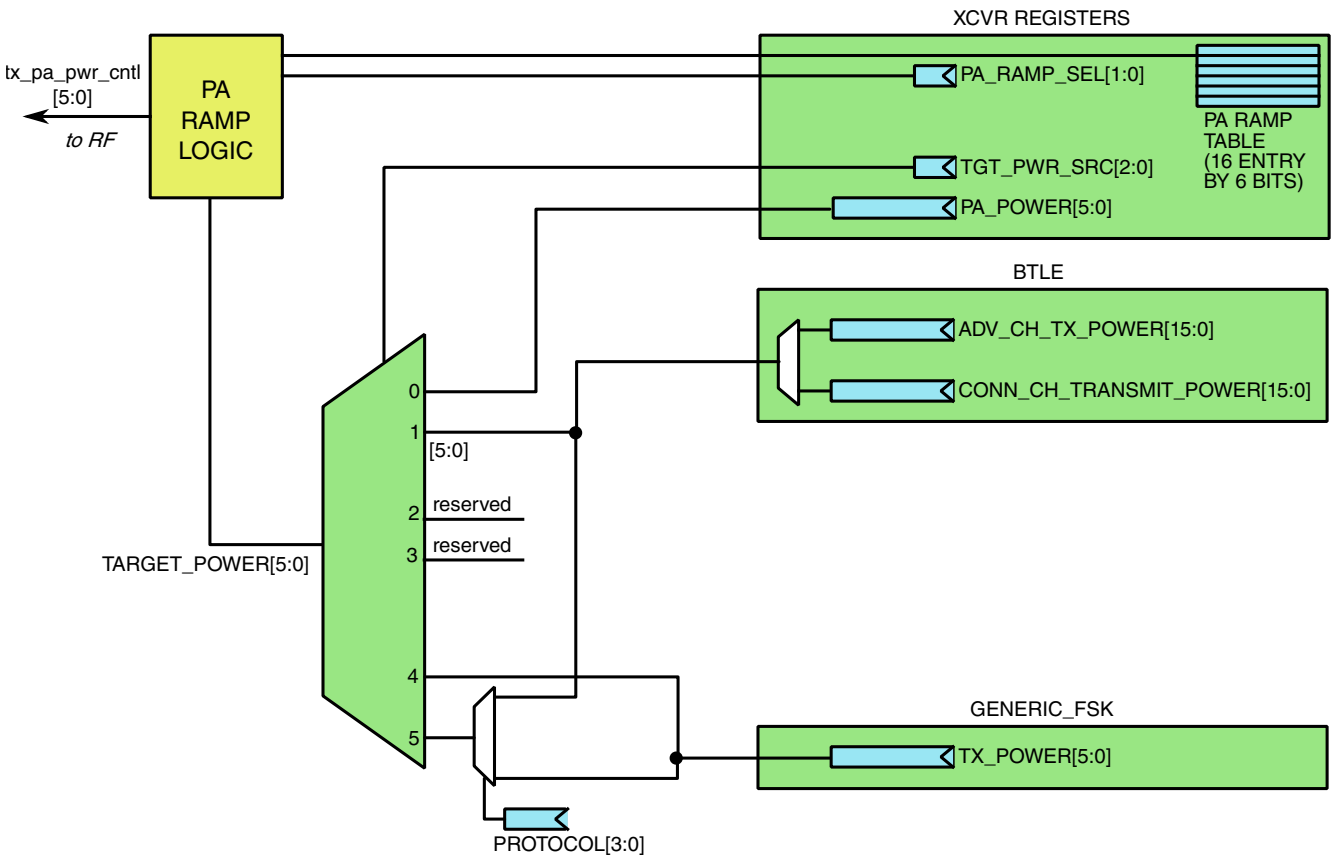


Figure 45-61. PA Target Power Selection

The XCVR register TGT_PWR_SRC[2:0], controls target power selection, according to the following table.

TGT_PWR_SRC[2:0]	TARGET POWER SOURCE
000	PA_POWER[5:0] register (XCVR space)
001	BTLE Link Layer
011	Reserved
100	GENERIC_FSK Link Layer (TX_POWER[5:0] register in GENERIC space)
101	Reserved
110	Reserved
111	PROTOCOL[3:0] bits select target power source

These bits reside in the XCVR_CTRL register in XCVR address space.

See Section “BTLE Radio Operation” for a description of how the BTLE Link Layer controls target power.

The TSM provides power control information to the PA as a 6-bit output: tx_pa_pwr_ctrl[5:0].

45.3.5.2.11 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 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 1, 2, or 4µs (programmable).

During a TX sequence, PA ramping, if enabled, is triggered by a low-to-high transition on the TSM output **tx_pa_en**, which is also the enable for the PA (tx buffer). At assertion of **tx_pa_en** during the TX sequence with ramping enabled, the TSM will enable the PA, with a minimum power level determined by the register PA_RAMP0[5:0]. After that, power level will increment according to a 16-deep pre-programmed register table (PA_RAMP 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_RAMP Table, from 0-15. After the 16th and final ramp step, PA power level will be the selected target power. The target power will remain in effect for the duration of the packet transmission, i.e., the "ON" phase of the TSM seqeuence.

The ramp trajectory is determined by the contents of the PA_RAMP Table. The PA_RAMP table contains register fields PA_RAMP0[5:0] … PA_RAMP15[5:0]. These register fields can be found in the PA_RAMP_TBL0, PA_RAMP_TBL1, PA_RAMP_TBL2, and PA_RAMP_TBL3 registers in XCVR address space. The default contents of the PA_RAMP table are as shown:

PA_RAMP Table Entry	POWER CONTROL VALUE (DEFAULT)
PA_RAMP0[5:0]	0x01
PA_RAMP1[5:0]	0x02
PA_RAMP2[5:0]	0x04
PA_RAMP3[5:0]	0x06
PA_RAMP4[5:0]	0x08
PA_RAMP5[5:0]	0x0C
PA_RAMP6[5:0]	0x10
PA_RAMP7[5:0]	0x14
PA_RAMP8[5:0]	0x18
PA_RAMP9[5:0]	0x1C
PA_RAMP10[5:0]	0x22
PA_RAMP11[5:0]	0x28
PA_RAMP12[5:0]	0x2C
PA_RAMP13[5:0]	0x30
PA_RAMP14[5:0]	0x36
PA_RAMP15[5:0]	0x3C

At each step of the ramp-up, the selected entry of the PA_RAMP Table is compared against target_power. The lesser of the two becomes the output power for that ramp step. This allows the same PA_RAMP table to be used for various target_power levels, without re-programming the table.

The ramp rate is controlled by the PA_RAMP_SEL[1:0] field 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 (32MHZ)	DURATION PER RAMP STEP (32MHZ)	TOTAL RAMP DURATION (26MHZ)	DURATION PER RAMP STEP (26MHZ)
00	No ramp	No ramp	No ramp	No ramp
01	1.0µs	2 Reference Clocks (0.0625µs)	1.23µs	2 Reference Clocks (0.077µs)
10	2.0µs	4 Reference Clocks (0.125µs)	2.46µs	4 Reference Clocks (0.154µs)
11	4.0µs	8 Reference Clocks (0.25µs)	4.92µs	8 Reference Clocks (0.308µs)

When PA_RAMP_SEL[1:0] is set to 0, there is no ramping, and tx_pwr_ctrl[5:0] tracks the selected Target Power at all times (see Section PA Target Power).

When PA_RAMP_SEL[1:0] is set to 1, a 1µs ramp is generated, as shown:

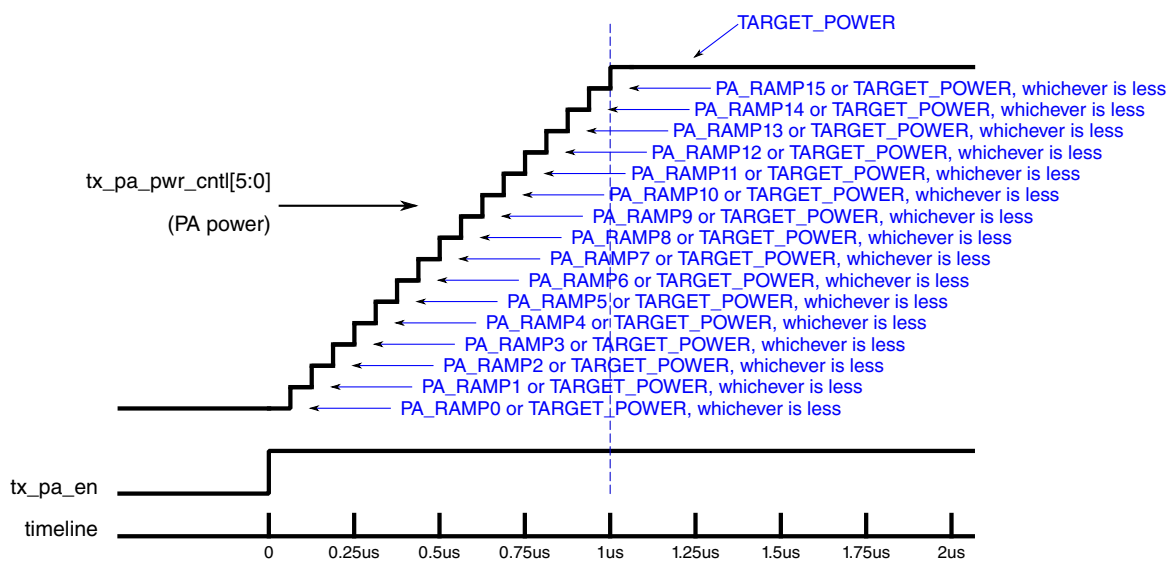


Figure 45-62. 1µS PA RAMP UP (32MHZ TIMING)

When PA_RAMP_SEL[1:0] is set to 2, a 2µs ramp is generated, as shown:

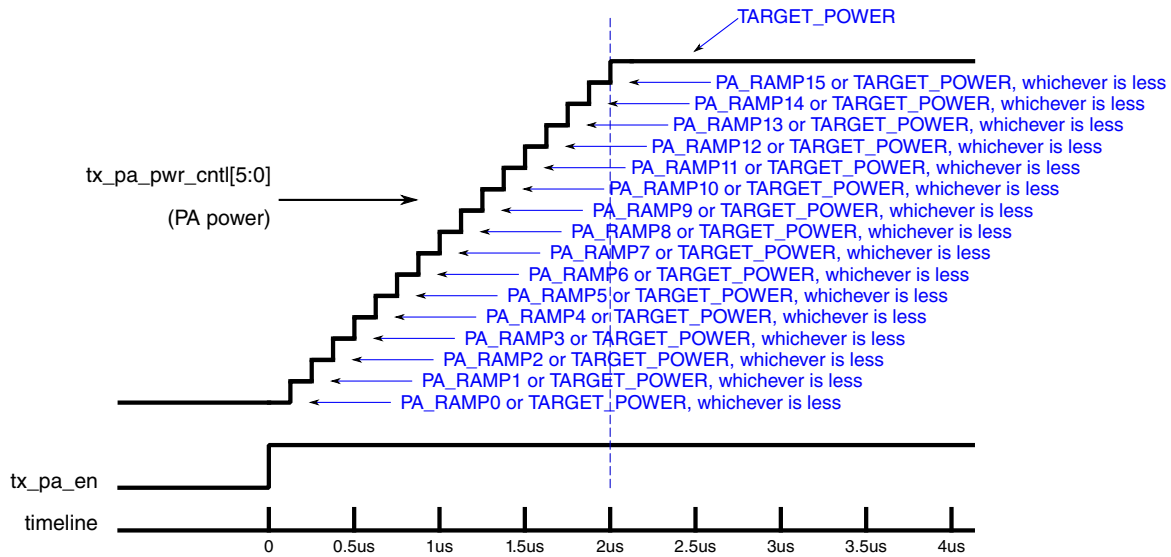


Figure 45-63. 2µS PA RAMP UP (32MHZ TIMING)

When PA_RAMP_SEL[1:0] is set to 3, a 4µs ramp is generated, as shown:

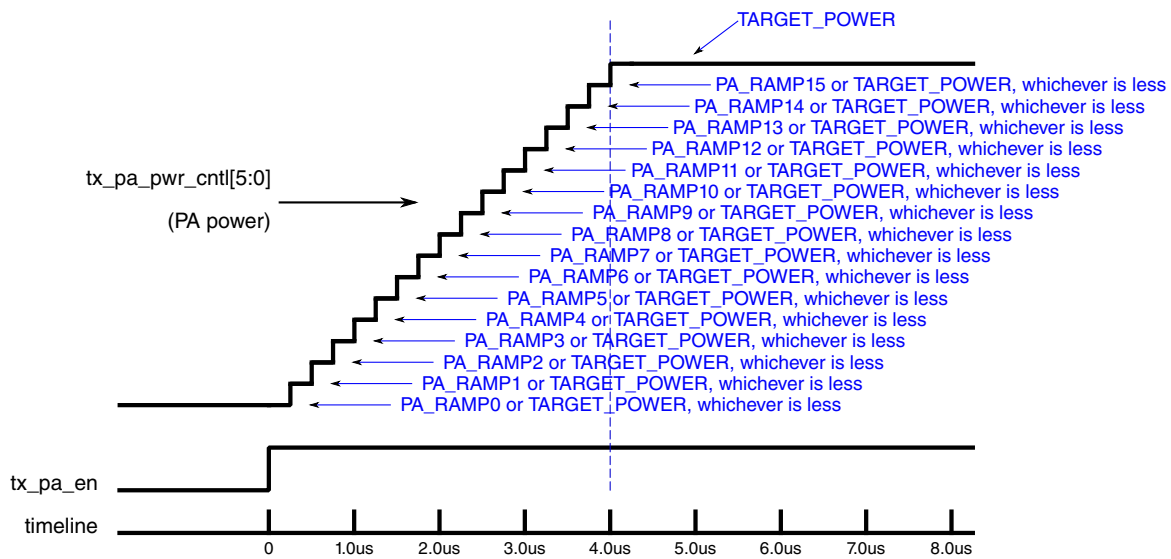


Figure 45-64. 4µS PA RAMP UP (32MHZ TIMING)

PA ramp-down is the mirror image of ramp-up. Ramp-down is triggered by the deassertion of the sequence-initiating event (e.g., btle_tx_en, or force_tx_en). At deassertion of the initiating event, output power will continue to hold at target_power for a programmable duration, controlled by the register TSM_CTRL[RAMP_DN_DLY]. This 4-bit register accommodates filter, and other, delay in the TX path before beginning the ramp down. RAMP_DN_DLY[3:0] is in units of µs. The default for RAMP_DN_DLY is 4. The effect of RAMP_DN_DLY, can be viewed as extending the "ON" phase of the TSM by a programmable number of microseconds. After

RAMP_DN_DELAY expires, output power will step through the PA_RAMP Table in reverse order, starting at PA_RAMP15, and ending at PA_RAMP0. At each step of the ramp-down, the selected entry of the PA_RAMP 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_RAMP0 level), **tx_pa_en** deasserts to disable the PA.

The following diagram shows an example of PA ramp down. In this example, PA_RAMP_SEL=2 (2µs), and RAMP_DN_DLY=3 (3µs):

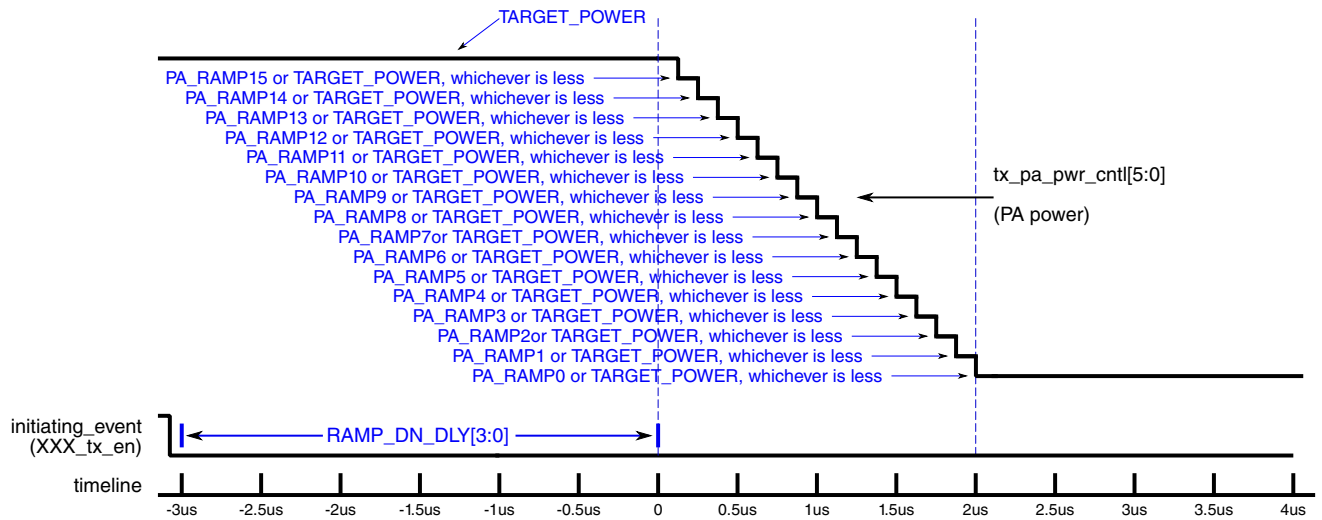


Figure 45-65. PA RAMP DOWN (32MHZ TIMING)

45.3.5.2.12 Front End Module (FEM)

FEM COMMON PIN INTERFACE

The FEM Common Pin Interface consists of 2 pins: TX_SWITCH, and RX_SWITCH. Both pins are outputs from the radio. The switch outputs convey timing information which can be used to control an external LNA or PA, via use of a Front End Module, or FEM. The FEM Common Pin Interface offers 2 configurations that allow for a range of FEM control schemes. These configurations are known as single mode and dual mode.

Several registers are provided to control the operation of the FEM interface, described in the table below.

FEM INTERFACE REGISTERS

Table 45-34. FEM Registers

Field	R/W	Description
ANTX_CTRLMODE	rw	**FEM control mode-** selects the single or dual mode **0 : single mode.** TXSWITCH = gpio2_trig_en RXSWITCH= (gpio2_trig_en OR gpio3_trig_en) **1:dual mode:** TX_SWITCH = gpio2_trig_en RX_SWITCH = gpio3_trig_en
ANTX_POL[3:0]	rw	**Antenna controls mode-** Control the polarity of the FEM interface pins: ANTX_POL[2]=1 : invert the TX_SWITCH output ANTX_POL[3]=1 : invert the RX_SWITCH output **NOTE:** ANTX_POL[0] and ANTX_POL[1] currently have no functionality.
FAD_NOT_GPIO[3:0]	rw	This register should be programmed to 0000 for FEM interface operation

Note: All register bits and fields listed in the table above, reside in the FAD_CTRL register in XCVR space.

In TSM mode (FAD_NOT_GPIO[2]=0), TX_SWITCH output timing is derived from TSM output **gpio2_trig_en**. Program TSM_TIMING49 register to obtain the desired timing on the TX_SWITCH pin. TX_SWITCH signalling is affected by the ANTX_CTRLMODE and ANTX_POL[2] settings.

In TSM mode (FAD_NOT_GPIO[3]=0), RX_SWITCH output timing is derived from TSM output **gpio3_trig_en**. Program TSM_TIMING50 register to obtain the desired timing on the RX_SWITCH pin. RX_SWITCH signalling is affected by the ANTX_CTRLMODE and ANTX_POL[3] settings.

The following diagram illustrates the FEM pinout and register controls.

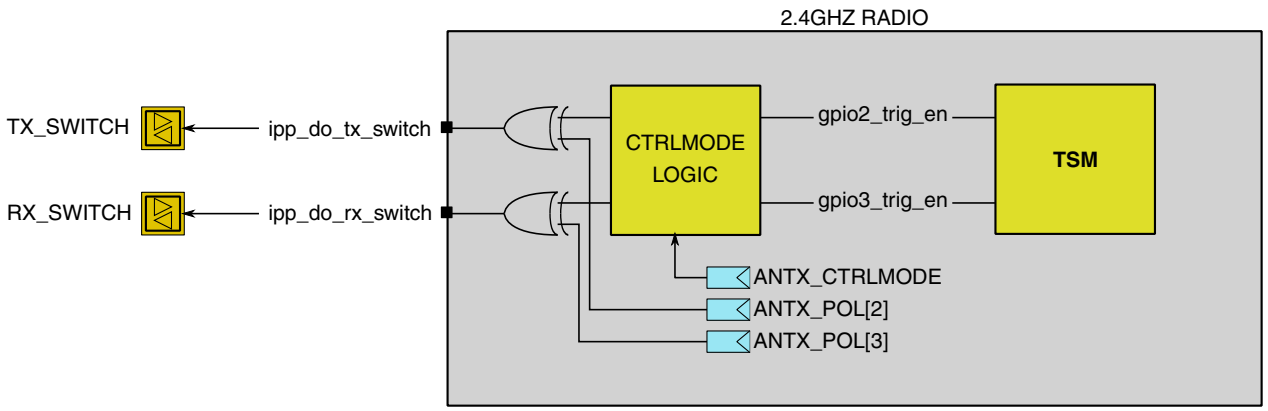


Figure 45-66. FEM Common Pin Interface

FRONT END MODULE

In 2.4GHz systems, the Radio Frequency (RF) Front-End Module is one of the most critical parts. Acting as an interface between the antenna and RF transceiver, the RF front-end module, or FEM, includes sensitive components such as an antenna Low Noise Amplifier (LNA) , power amplifiers (PA), antenna tuning switches, and a power management unit.

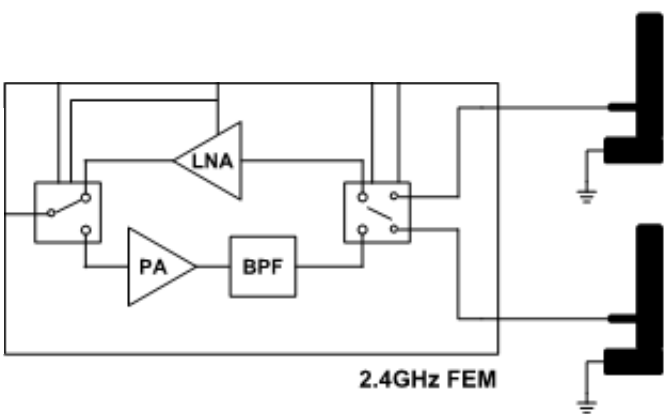


Figure 45-67. FEM Block Diagram

The 2.4GHz radio does not incorporate an antenna-selection engine, and therefore has no antenna-selection outputs to drive the FEM module. However the radio does include TX_SWITCH and RX_SWITCH outputs, with programmable timing, with which to control the external PA and/or external LNA components of the FEM.

To accommodate multiple FEM interfaces and control schemes, the radio's FEM module incorporates 2 Pin Control Modes, which are governed by the register control bit ANTX_CTRLMODE.

Some FEM devices use a single input pin to switch the antenna selection. This is called single-mode operation. Although the radio does not provide antenna-selection outputs, single-mode operation does affect the timing signals presented on the TX_SWITCH and RX_SWITCH pins. Operate the FEM module in this mode by programming ANTX_CTRLMODE=0. A timing diagram of single-mode operation is shown below. In this example, two packets are received back-to-back, and both received packets require an auto-Acknowledge packet to be transmitted immmediately following reception.

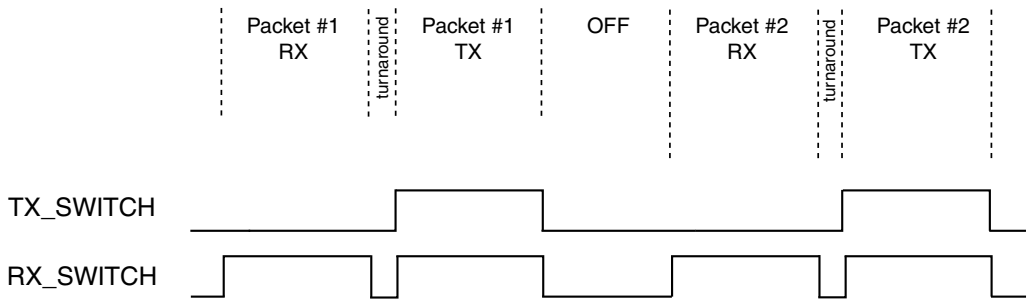


Figure 45-68. FEM Single Mode Timing

An example FEM Interface diagram, operated in single mode, is shown below.

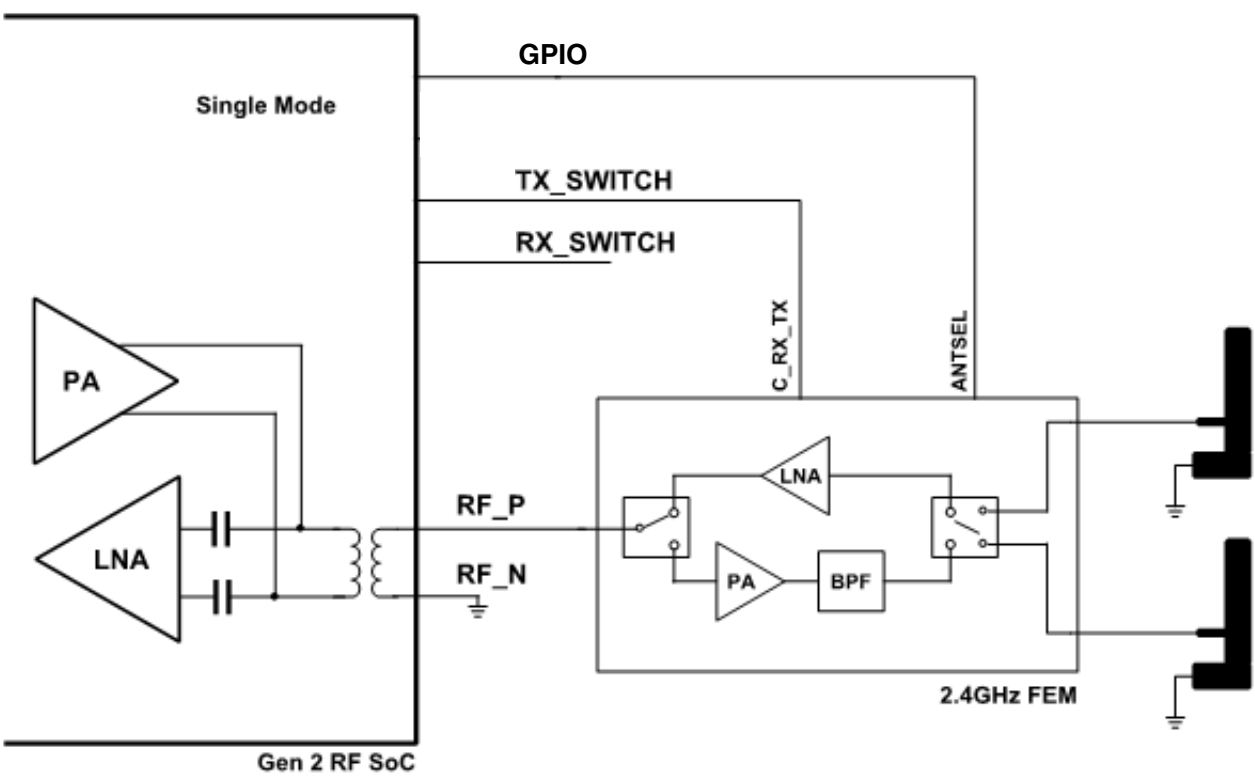


Figure 45-69. FEM Interface Single Mode

Other FEM devices use 2 input pins to switch the antenna selection. This is called dual-mode operation. Operate the FEM module in this mode by programming ANTX_CTRLMODE=1. A timing diagram of dual-mode operation is shown below. In this example, two packets are received back-to-back, and both received packets require an auto-Acknowledge packet to be transmitted immmediately following reception.

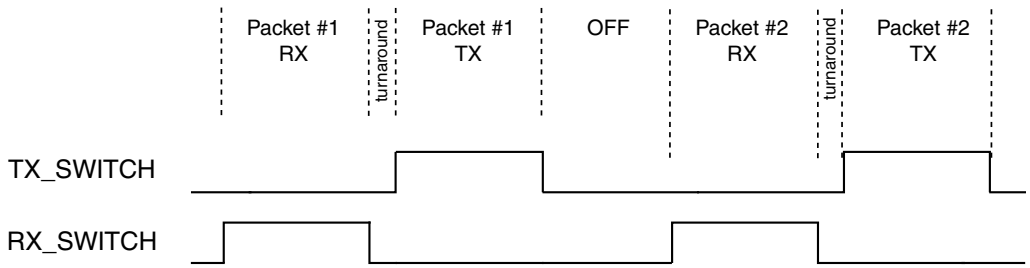


Figure 45-70. FEM Dual Mode Timing

An example FEM Interface diagram, operated in dual mode, is shown below.

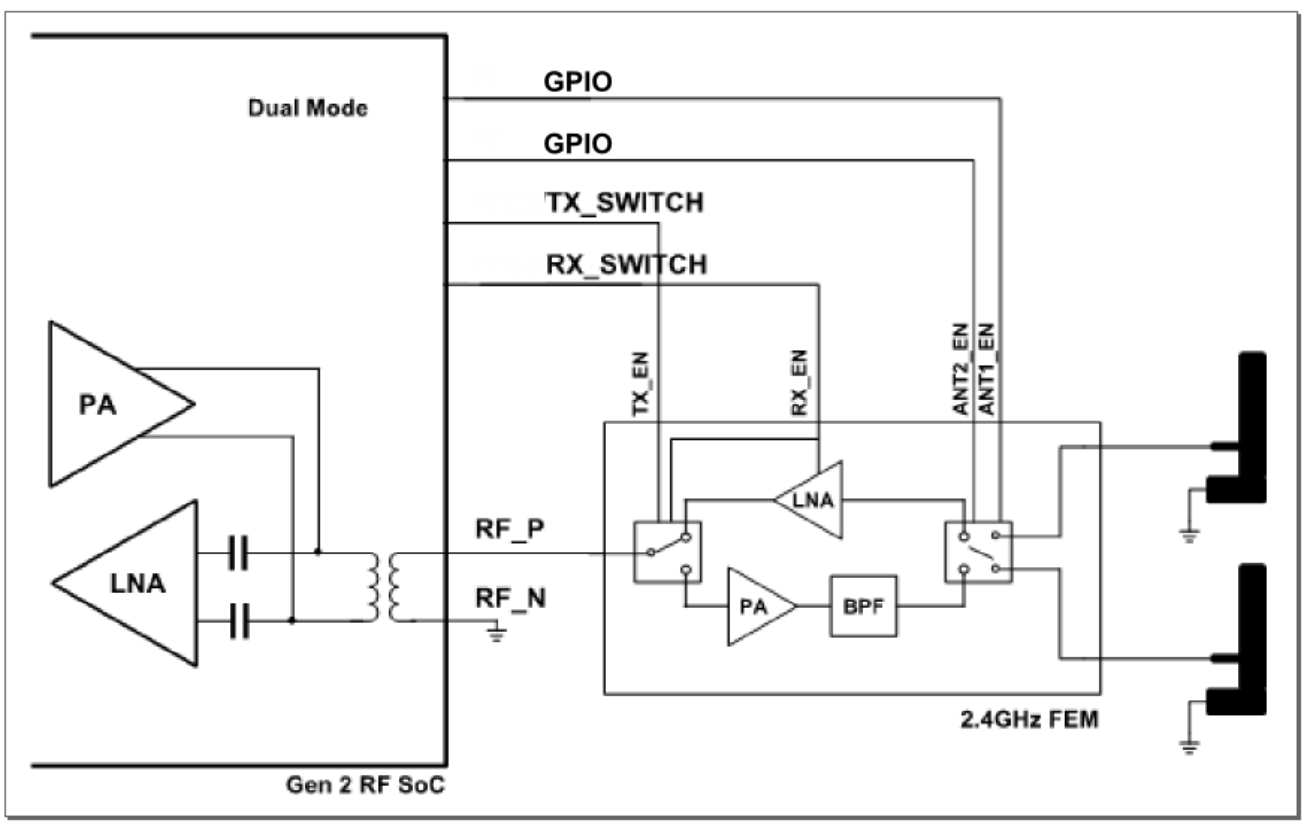


Figure 45-71. FEM Interface Dual Mode

Other FEM use cases, include external PA only, external LNA only, or external PA plus LNA. In such cases, the ANT_A and ANT_B inputs to the FEM can be tied off (not controlled by radio), and TX_SWITCH and RX_SWITCH pins can be operated in TSM mode (pure timing signals). This is called Single Antenna configuration. An example FEM interface diagram illustrating such a use case, is shown below.

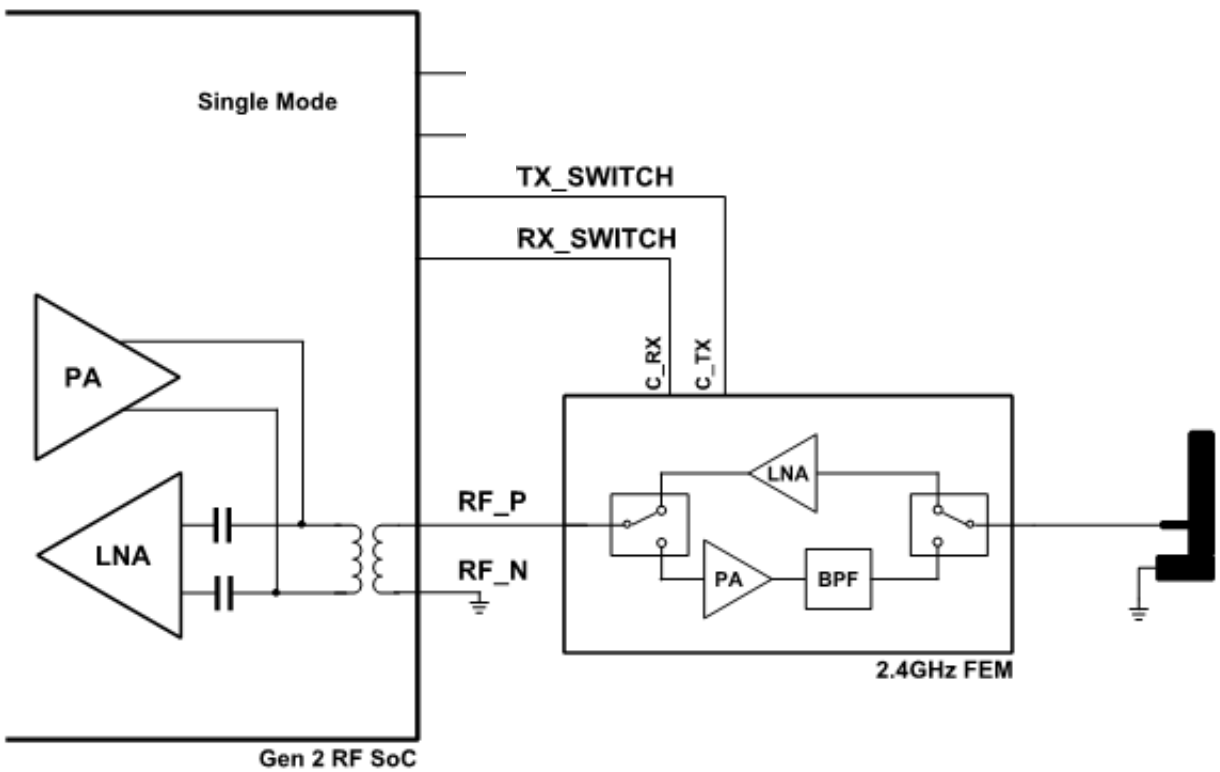


Figure 45-72. FEM Interface Single Antenna Configuration

45.3.5.2.13 TSM Fast Warmups

The previous sections illustrate the detailed control and timing generated by the TSM, operating in nominal-timing mode. There are situations that allow for the possibility to streamline and shorten the TSM warmup sequences. These situations arise when the transceiver is operating on the same RF channel for many consecutive transmit and/or receive operations. Multiple closely-spaced TX operations on the same channel allow the HPM calibration to be skipped, after such a calibration is performed once on a given RF channel; similarly, multiple closely spaced RX operations on the same channel allow DC offset calibration to be skipped, after such a calibration is performed once on a given RF channel.

Fast TSM warmups have been devised to accomplish this streamlining. There is one fast warmup defined for TX, and one for RX. Once engaged, during a fast warmup, a portion of the nominal, programmed warmup is bypassed, or skipped over, by instructing the TSM counter to jump, once it reaches a certain count, to a new count that is not just an increment-by-one. The starting point for the “jump” as well as the destination count, or end point for the “jump”, are TSM programmable registers. There are one pair of start/ destination registers for TX, and another for RX, allowing one such jump per sequence. In addition, enable bits provide software with fine control over precisely when fast warmups are allowed, versus nominal warmups.

Interaction with other blocks is required to support TSM fast warmup capability. The PLL digital block is responsible for selecting the correct frequency for RF operations, and will keep track of whether consecutive warmup operations utilize the same channel or not; it will signal this information to the TSM. The PLL digital block maintains such tracking separately for TX and RX warmups. The TSM will then signal back to the PLL digital block when a TX warmup is actually “fast” or not; the PLL digital block will use this information to instruct its state machine to skip HPM calibration. For RX warmups, the TSM will signal to the RX digital block, when to skip DC calibration, and instead use stored values.

For non-BLE protocols, channel selection is a purely software function. For these protocols (e.g. GENERIC_FSK), a CHANNEL_NUM register is programmed to select the RF channel for the next TX or RX warmup. For these protocols, any change to the respective CHANNEL_NUM register will erase the "channel memory" that the PLL Digital block keeps track of determine whether the next TSM warmup is allowed to be a fast, rather than a normal warmup. Any write to CHANNEL_NUM will force the next TX warmup to be normal (not fast), regardless of whether the *previous* TX warmup was on the same channel or not. Similarly, any write to CHANNEL_NUM will force the next RX warmup to be normal (not fast), regardless of whether the *previous* RX warmup was on the same channel or not.

The inter-block signaling to support fast TSM warmups is illustrated in the following diagram, followed by a table which describes the signals.

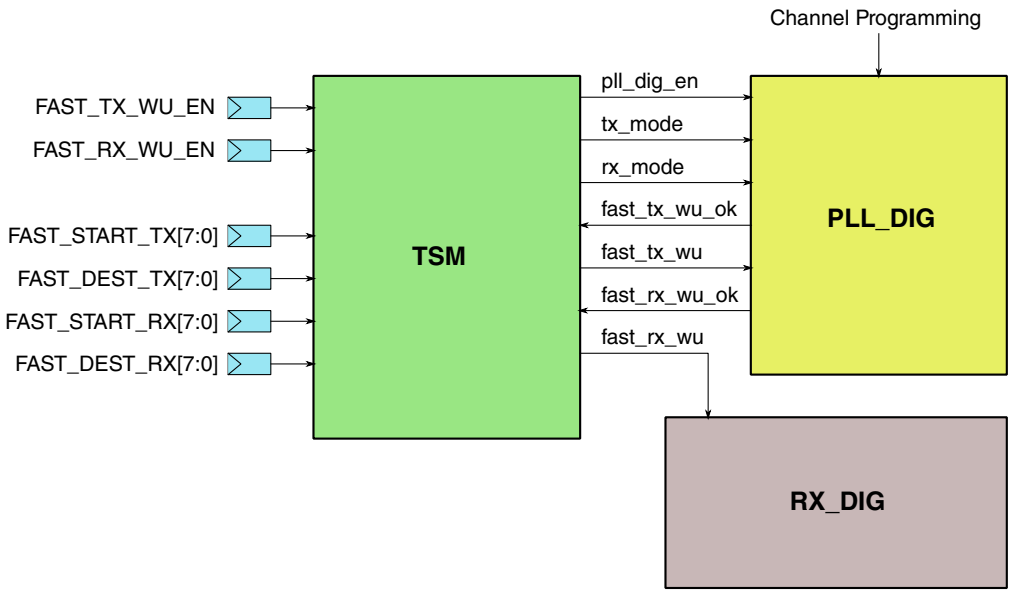


Figure 45-73. Transceiver Signalling To Support TSM Fast Warmups

"Fast Warmup" Signal	Source	Destination	Description
pll_dig_en	TSM	PLL Digital	Rising edge of this signal used by PLL_DIG to capture the Channel Programming for the current warmup
tx_mode	TSM	PLL Digital	Used in conjunction with pll_dig_en to capture the Channel Programming for TX sequences
rx_mode	TSM	PLL Digital	Used in conjunction with pll_dig_en to capture the Channel Programming for RX sequences
fast_tx_wu_ok	PLL Digital	TSM	Indicates that the Channel Programming on the *current* TX warmup has not changed since the *last* TX warmup
fast_tx_wu	TSM	PLL Digital	This is a fast TX warmup. PLL_DIG should skip HPM Cal.
fast_rx_wu_ok	PLL Digital	TSM	Indicates that the Channel Programming on the *current* RX warmup has not changed since the *last* RX warmup
fast_rx_wu	TSM	RX Digital	This is a fast RX warmup. RX Dig should skip DCOC Cal.

TSM Registers to support Fast TSM Warmups are described in the following table.

Field	R/W	Description
FAST_TX_WU_EN	rw	1: Enable Fast TX Warmup (if fast_tx_wu_ok asserted by PLL_DIG) 0: Disable Fast TX Warmups
FAST_RX_WU_EN	rw	1: Enable Fast RX Warmup (if fast_rx_wu_ok asserted by PLL_DIG) 0: Disable Fast RX Warmups
FAST_WU_CLEAR	rw	Erase the "channel memory" in the PLL Digital Block, forcing the next TX warmup and RX warmup to be normal warmups (not fast). This bit is not self-clearing. Write a '1' to the bit, followed by '0', to erase channel history.
FAST_START_TX[7:0]	rw	During a Fast TX Warmup, this value represents the starting TSM count for the portion of the warmup that is to be skipped over. If Fast TX Warmups are enabled, this count will never be reached; its place in the TSM warmup will be replaced by the contents of FAST_DEST_TX[7:0], thereby executing the jump.
FAST_DEST_TX[7:0]	rw	During a Fast TX Warmup, this value represents the ending TSM count for the portion of the warmup that is to be skipped over. If Fast TX Warmups are enabled, FAST_START_TX[7:0] will never be reached and will instead be replaced by the contents of this register, thereby executing the jump. Example: FAST_START_TX = 10 FAST_DEST_TX = 15 The TSM will count as follows: … 7,8,9,15,16,17 …
FAST_START_RX[7:0]	rw	During a Fast RX Warmup, this value represents the starting TSM count for the portion of the warmup that is to be skipped over. If Fast RX Warmups are enabled, this count will never be reached; its place in the TSM warmup will be replaced by the contents of FAST_DEST_RX[7:0], thereby executing the jump.
FAST_DEST_RX[7:0]	rw	During a Fast RX Warmup, this value represents the ending TSM count for the portion of the warmup that is to be skipped over. If Fast RX Warmups are enabled, FAST_START_RX[7:0] will never be reached and will instead be replaced by the contents of this register, thereby executing the jump. Example: FAST_START_TX = 10 FAST_DEST_TX = 15 The TSM will count as follows: … 7,8,9,15,16,17 …

Software running on the various protocol engines which utilize the TSM to provide access to the channel, need not be aware that the total warmup time has been reduced, when the TSM executes a fast warmup, as opposed to a nominal warmup. Link Layer software expects TSM warmup times to be a fixed constant, actually 2 fixed constants, one for TX warmups and one for RX warmups. Software must account for these TX and RX warmup times when they are scheduling transceiver operations, so that the first symbol of preamble is transmitted at the correct instant for transmit operations, and that

the receiver is ready to receive the first symbol of preamble at the correct instant for receive operations. Software views these warmup times as fixed lead times that must be accounted for in advance of a transceiver operation.

In order to make the fast warmup time reduction transparent to software, when a Fast TSM warmup is executed, the TSM will delay the start of the warmup, by the amount of time skipped by the jump. That keeps the total elapsed time, from the instant the software launches the TSM sequence, to the point at which the warmup completes, identical for fast and nominal warmups.

The following diagram illustrates how the TSM delays the start of the Fast TX sequence, relative to the nominal TX sequence, so that the delta time between btle_tx_en (TSM launch signal), and the transmission of the first bit of preamble, is the same in either case.

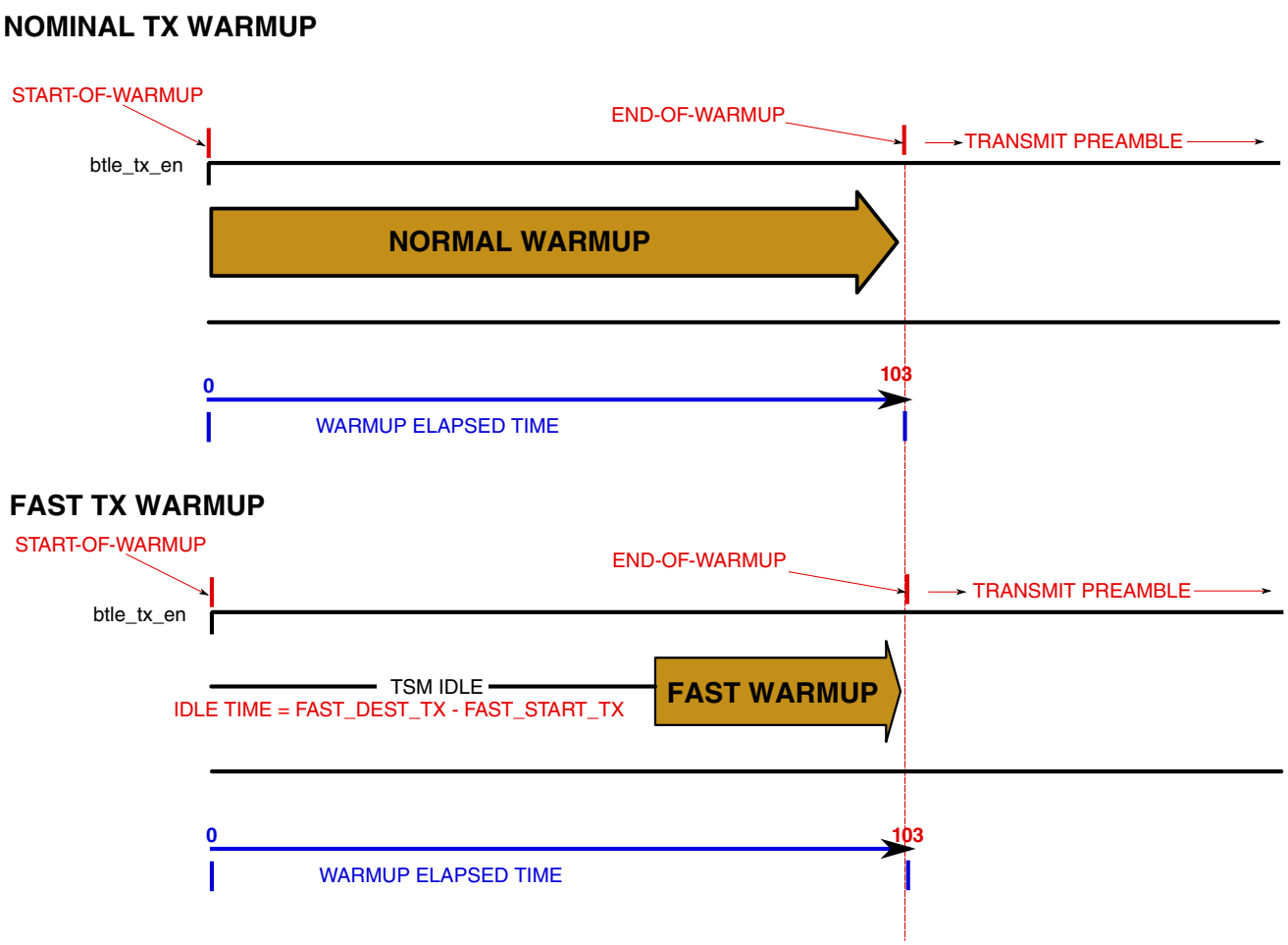


Figure 45-74. Nominal vs. Fast TX Warmups

The following diagram illustrates how the TSM delays the start of the Fast RX sequence, relative to the nominal RX sequence, so that the delta time between btle_rx_en (TSM launch signal), and the ability to receive first bit of preamble, is the same in either case.

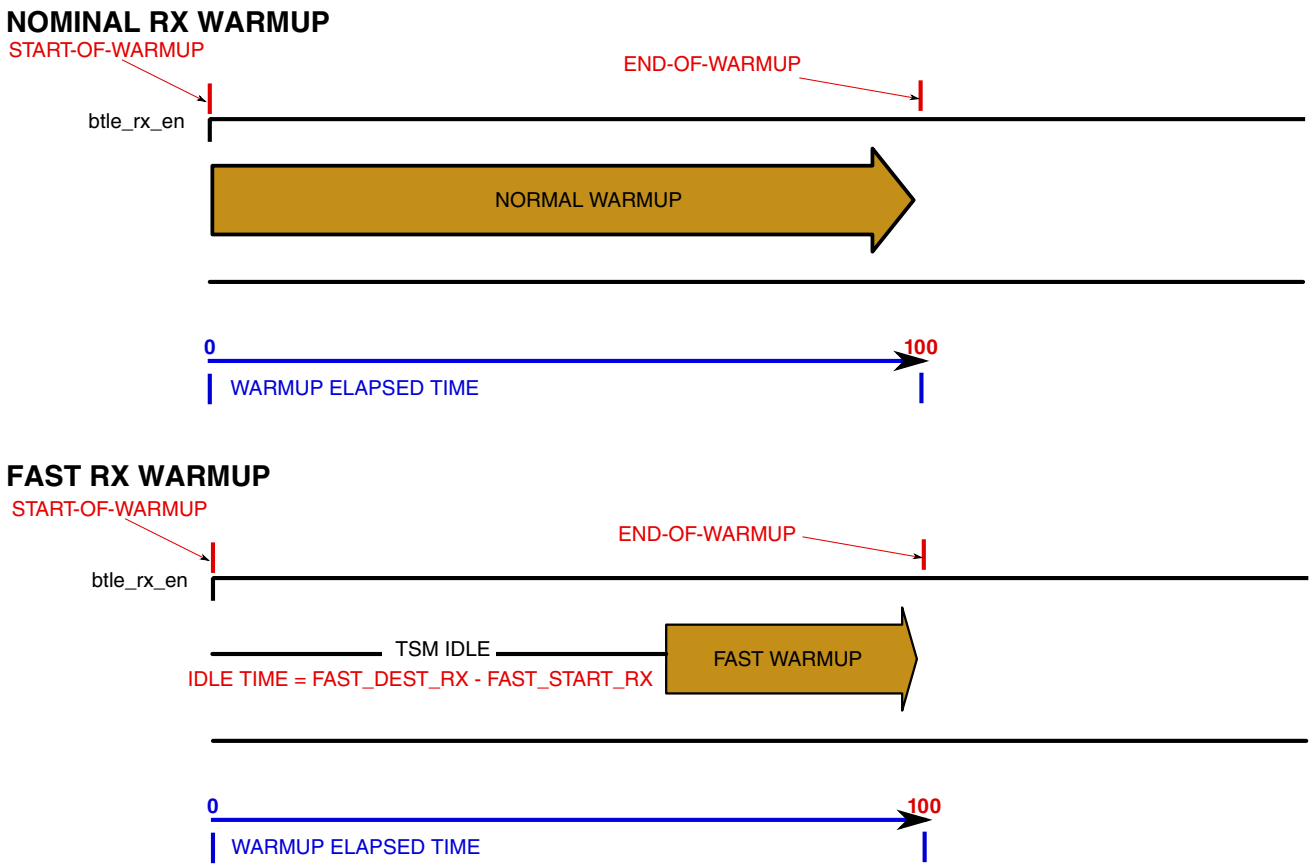


Figure 45-75. Nominal vs. Fast RX Warmups

In the previous examples, a BLE-triggered warmup is shown for TX and RX. However the same concept applies to all protocols which use the TSM to access the RF channel. The diagrams show a 103us TX warmup and a 100us RX warmup. The actual length-of-warmup is programmable, determined by the END_OF_SEQ register.

45.3.5.2.14 Coexistence

The 2.4GHz radio includes mechanisms which facilitate packet traffic arbitration (PTA) between two collocated radios that will be sharing the same antenna for the 2.4GHz ISM band communications. For such mechanisms, IEEE 802.15.2 proposes MAC layer techniques to coordinate scheduling of packet transmission between the two wireless (WLAN and WPAN) networks by implementation of local controls. The mechanisms discussed in this document can be easily extended to work with the Bluetooth-LE radio, and in fact, all protocols currently supported by the 2.4GHz radio.

One such mechanism places the PTA arbiter on the WLAN device, and envisions a 3-wire interface to/from the WPAN device to the PTA. This protocol is supported by BT devices from some third-party vendors (Broadcom, CSR), and is shown in the image below ("BLUETOOTH" can be extrapolated to include any of the protocols supported by the 2.4GHz WPAN radio).

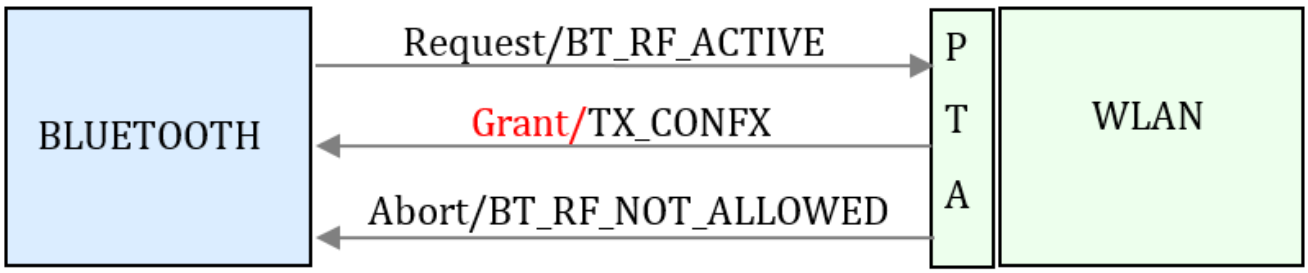


Figure 45-76. 3-WIRE THIRD PARTY INTERFACE

An alternative is the Nokia 4-wire interface, which is shown in the image below. (A 3-wire variant of the Nokia option exists, which merely omits the FREQ signal):

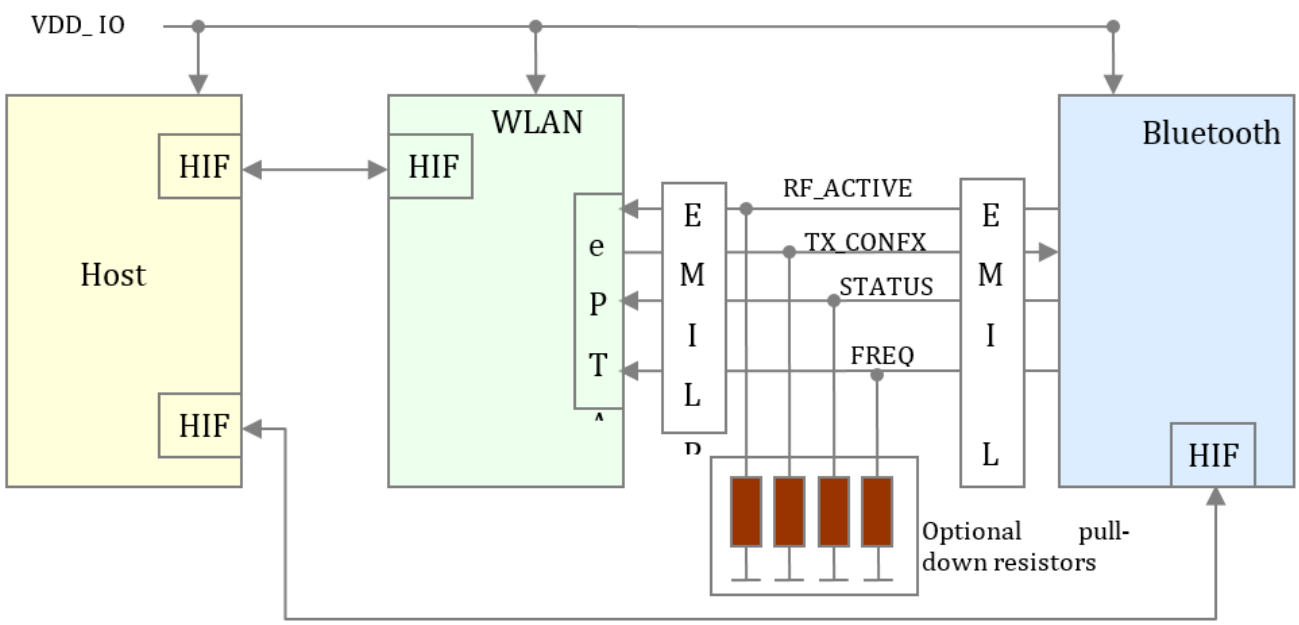


Figure 45-77. 4-WIRE THIRD PARTY INTERFACE

In either case, the signal RF_ACTIVE is asserted prior to any 2.4GHz WPAN radio transaction and it remains active for the duration of the transaction. Transaction is defined as a polling slot(s) and an immediately following response slot(s). Retransmissions are treated as separate transactions. The RF_ACTIVE pin at the SoC level can be driven by

either an RSIM_RF_ACTIVE or a TSM_RF_ACTIVE source. TSM_RF_ACTIVE, since it is controlled by TSM, can be asserted at most ~200us before a radio event. TSM output TSM_SPARE1_EN is to be used as the TSM source for this purpose.

RF_ACTIVE needs to be asserted high a programmable amount of time before a radio event takes place and remains high until either the radio event has been completed or aborted. There should be no glitches/unwanted transitions between successive radio events (RX->TX or TX->TX or vice versa). This can be a problem for protocols, such as BLE which require the transmission of an acknowledge packet shortly after reception. These protocols also require reception of an acknowledge packet shortly after transmission. BLE Advertising and Scan states offer other scenarios where closely-spaced (but not adjoining) TX and RX operations are required. The inter-frame spacing between the successive RX and TX operations, is long enough such that TSM_SPARE1_EN will deassert and reassert between operations, creating the unwanted transitions which must be avoided.

The following timing diagram shows the desired (glitch-free) assertion of RF_ACTIVE, encompassing a BLE RX operation followed by a TX operation.

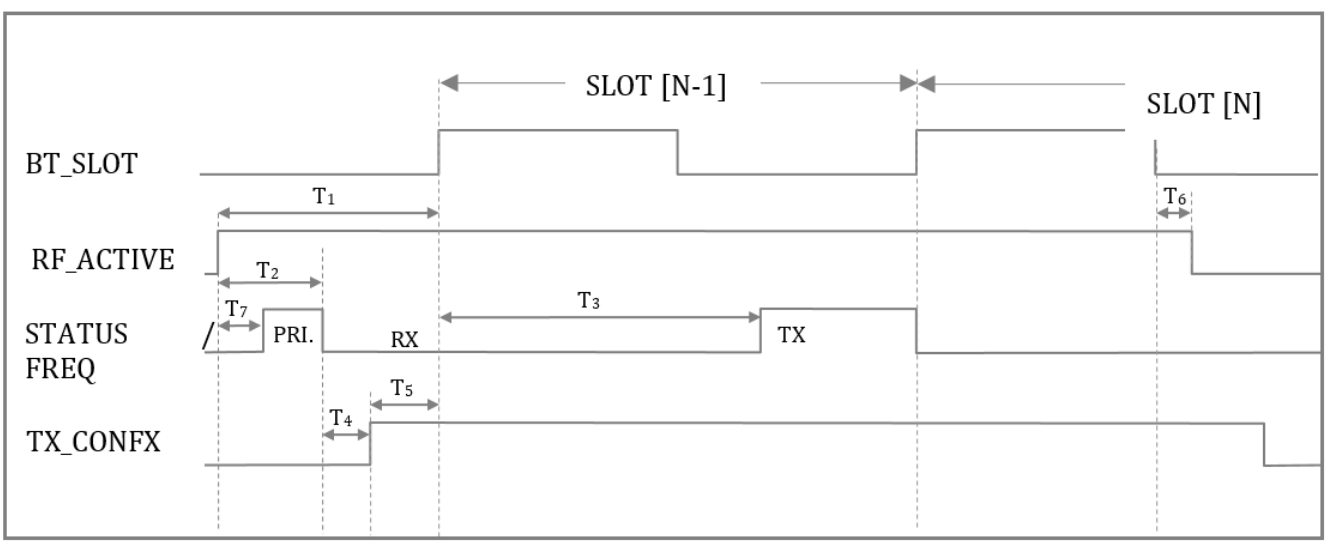


Figure 45-78. IDEAL RF_ACTIVE TIMING (NO GLITCH)

The following shows the actual signal timing which would occur on RF_ACTIVE (TSM_SPARE1_EN), in a typical BLE Advertising scenario, if no action were to be taken to address the unwanted, inter-frame transitions:

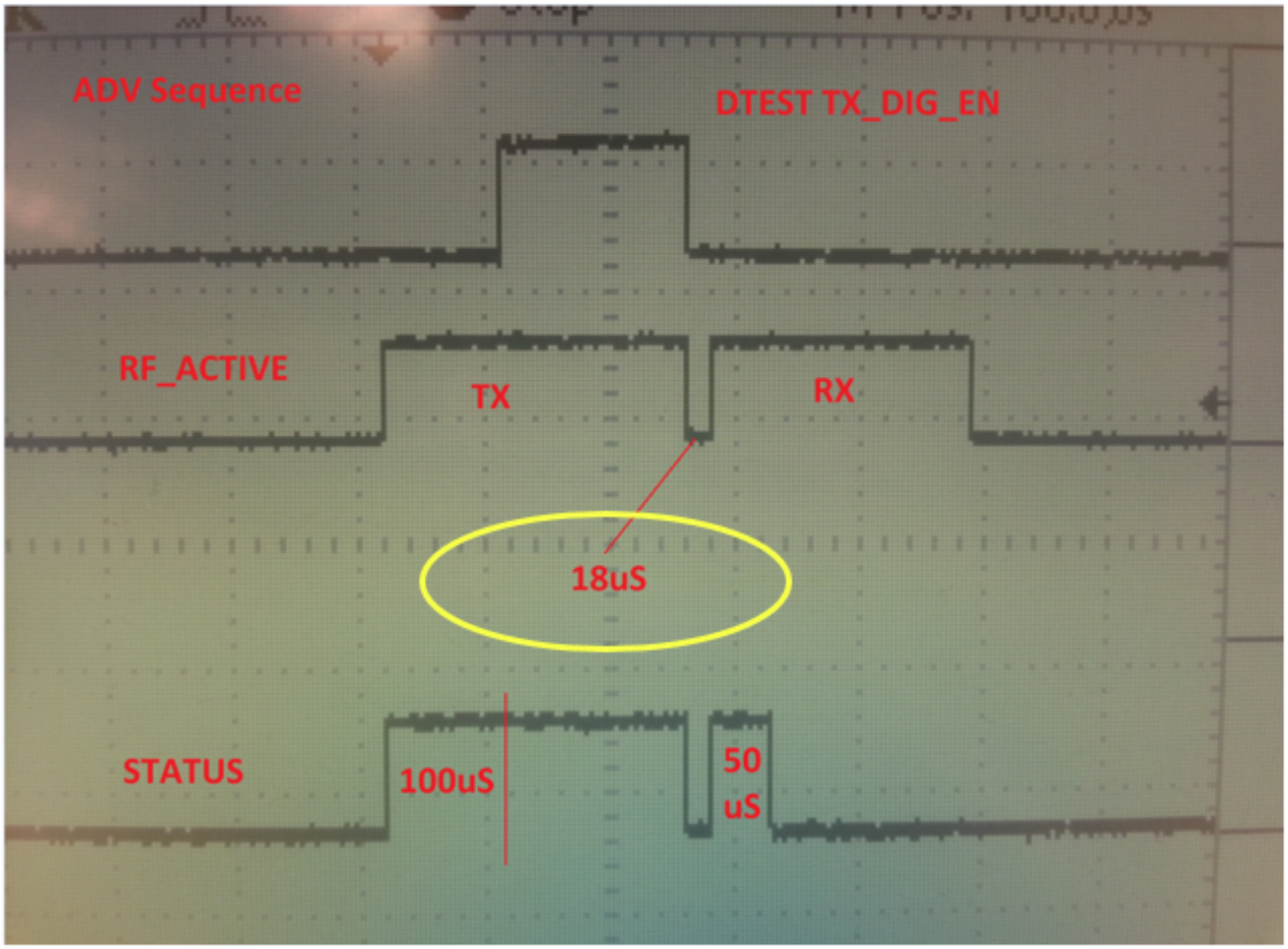


Figure 45-79. ACTUAL RF_ACTIVE TIMING (UNCOMPENSATED)

To remedy this, a feature has been added to TSM, to allow the assertion of TSM_SPARE1_EN output to be optionally extended by a programmable amount, after the nominal TSM sequence completes. The TSM_SPARE1_EN extension will occur for all sequences (TX and RX) for which TSM_SPARE1_EN is programmed to assert, whenever register TSM_SPARE1_EXTEND>0. The hardware mechanism for extending TSM_SPARE1_EN is illustrated in the following diagram:

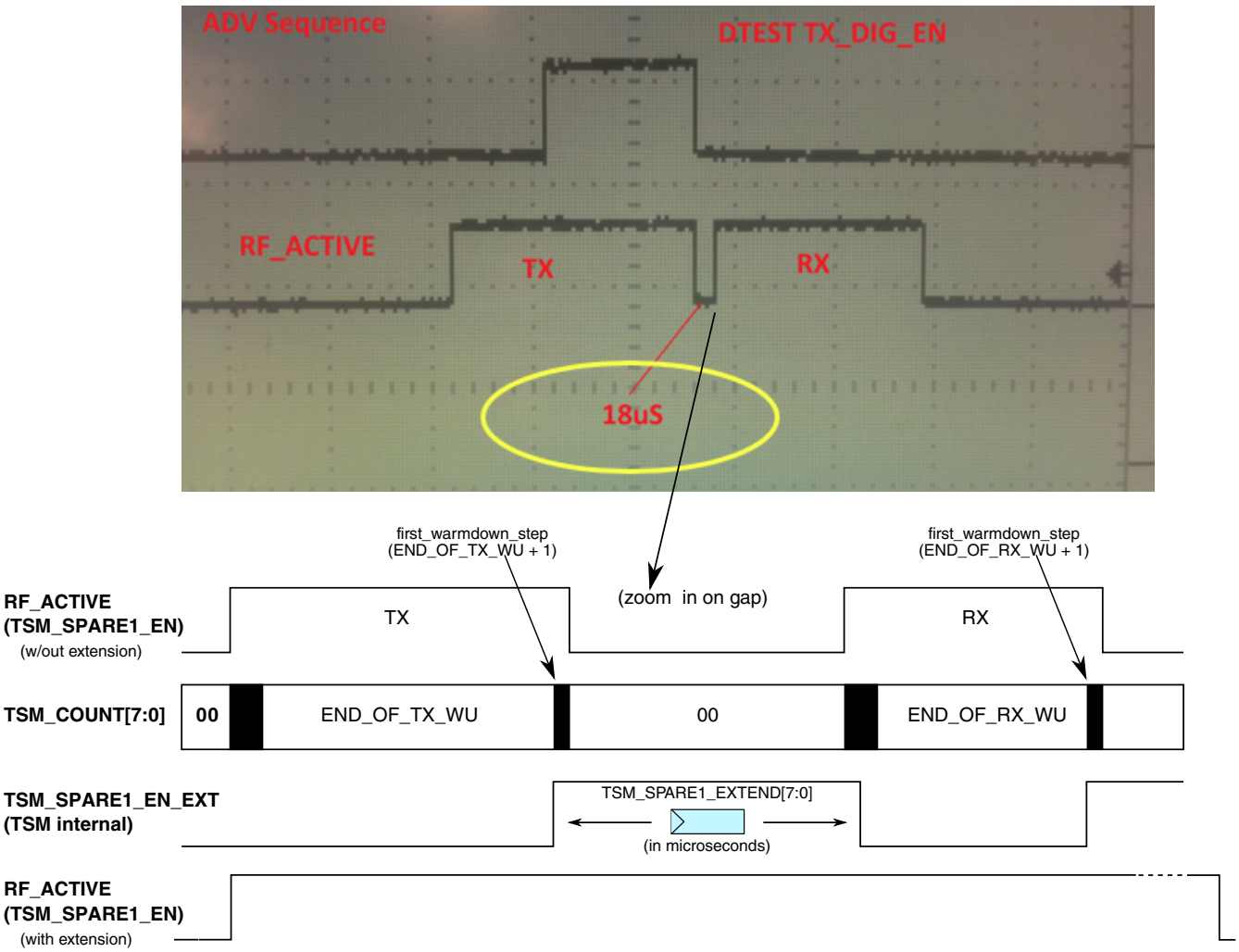


Figure 45-80. TSM_SPARE1_EN EXTENSION

The trigger for the extension is the TSM's first warmdown step (END_OF_xx_WU + 1), which will be reached whenever the Link Layer deasserts its TX/RX command, or on any abort. If enabled, the extension signal will assert (high) at this point, and will remain high for *TSM_SPARE1_EXTEND* microseconds. The extension signal (TSM_SPARE1_EN_EXT) is logically "OR-ed" with the existing TSM-controlled output TSM_SPARE1_EN, to generate the composite, extended RF_ACTIVE signal. Software can cut short the extension at any time by writing TSM_SPARE1_EXTEND=0. Setting this register to 0 also disables the extension feature from activating on any new TSM sequence. The register TSM_SPARE1_EXTEND[7:0] resides in the COEX_CTRL register of XCVR address space, and yields an extension range from 0 to 255µs.

45.3.5.2.15 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).

45.3.5.2.16 Clocks

The TSM uses 4 clocks:

1. ipg_clk
2. tsm_clk
3. tsm_dly_clk
4. tsm_4m_clk

The "**ipg_clk",** is a gated 32mhz (or 26mhz) clock from the reference oscillator. This clock only used to re-clock the tx_pwr_cntl[5:0] 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 67 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 (or 26mhz) source, the reference oscillator. They are divided down from this reference frequency in the CGM (Clocks Generator 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.

45.3.5.2.17 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.

45.3.5.2.18 Interrupts

For debug purposes, TSM can generate up to 2 interrupts to the host processor. Like all other TSM-controlled outputs, the point during the TX or RX warmup at which the interrupt is triggered, is fully programmable. Since there is no defined mission-mode use for TSM interrupts, no new dedicated timing registers have been provisioned. Instead, the 2 TSM interrupts share timing registers with TSM-controlled outputs **tsm_spare0_en** and **tsm_spare1_en**. The following diagram depicts the reuse scheme.

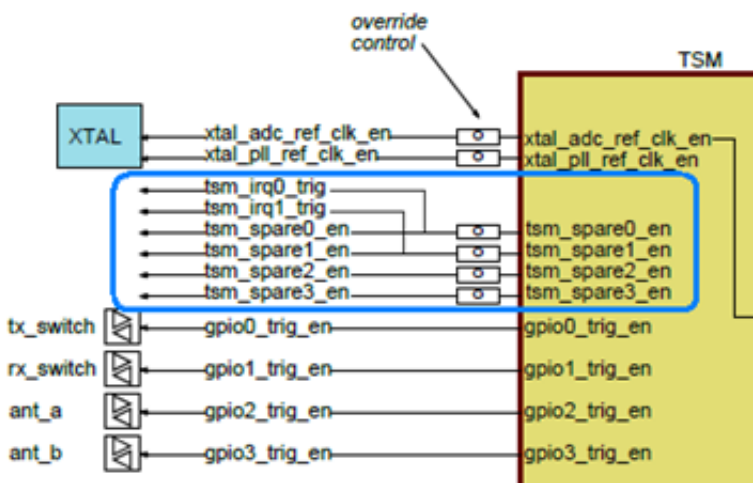


Figure 45-81. TSM Interrupt Re-use Scheme

Two new control bits have been added to TSM_CTRL to enable each TSM interrupt. The following table lists the TSM Interrupt, the TSM enable bit for the interrupt, and the TSM timing register which controls the timing of the interrupt.

Table 45-35. TSM Interrupt Control

TSM Interrupt	Interrupt Enable Bit (TSM_CTRL)	Timing Control Register
TSM_IRQ0	TSM_IRQ0_EN	TSM_TIMING43
TSM_IRQ1	TSM_IRQ1_EN	TSM_TIMING44

Because the TSM is part of XCVR space, and XCVR space does not have its own interrupt vector, the TSM interrupt is "assigned" to the Link Layer currently in possession of the 2.4GHz radio. Possession is determined by the state of the XCVR_CTRL[PROTOCOL] bits. Each of the Link Layer controllers (BLE, GENERIC_FSK), has in its address space, a TSM_IRQ status bit, as well as a enable bit to allow/disallow the TSM interrupt to assert the Link Layer's interrupt line to the MCU. The bit positioning of the TSM_IRQ varies slightly amongst the various Link Layers, to optimally group the TSM_IRQ with existing, protocol-specific interrupt sources, and to minimize the software impact to existing service routines, by requiring as few additional register reads as possible to service the Link Layer's interrupts with TSM_IRQ enabled. The TSM_IRQ status bit, as it appears in each Link Layer's address space, is a logical "OR" of the TSM's TSM0_IRQ and TSM1_IRQ. The Link Layer TSM_IRQ is read-only. It can be cleared by clearing both TSM0_IRQ and TSM1_IRQ in the XCVR's XCVR_STATUS register. In the XCVR_STATUS register, TSM0_IRQ and TSM1_IRQ are both write-1-to-clear bits. See the Register Description for each individual Link Layer, for a description of the TSM_IRQ read-only interrupt status bit, and its bit positioning within the Link Layer's address space.

45.3.5.3 Memory Map and register definition

There are 59 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 59 sets, the following table lists the SoC register name (TSM_TIMING00 – TSM_TIMING58), and 4 timing register names associated with each. There are various other registers to control the overall TX and RX sequence lengths, configure for Fast Warmups, etc.

See below for the TSM register locations and detailed definitions.

45.3.5.3.1 XCVR_TSM register descriptions

45.3.5.3.1.1 XCVR_TSM_ADDR Memory map

XCVR_TSM base address: 4005_C2C0h

Offset	Register	Width (In bits)	Access	Reset value
0h	TSM CONTROL (CTRL)	32	RW	FF00_4000h
4h	TSM END OF SEQUENCE (END_OF_SEQ)	32	RW	6766_6A63h
8h	PA POWER (PA_POWER)	32	RW	0000_0000h
Ch	PA RAMP TABLE 0 (PA_RAMP_TBL0)	32	RW	0604_0201h
10h	PA RAMP TABLE 1 (PA_RAMP_TBL1)	32	RW	1410_0C08h
14h	PA RAMP TABLE 2 (PA_RAMP_TBL2)	32	RW	2822_1C18h
18h	PA RAMP TABLE 3 (PA_RAMP_TBL3)	32	RW	3C36_302Ch
24h	TSM RECYCLE COUNT (RECYCLE_COUNT)	32	RW	001A_0464h
28h	TSM FAST WARMUP CONTROL 1 (FAST_CTRL1)	32	RW	0000_FF00h
2Ch	TSM FAST WARMUP CONTROL 2 (FAST_CTRL2)	32	RW	FFFF_FFFFh
30h	TSM_TIMING00 (TIMING00)	32	RW	6700_6A00h
34h	TSM_TIMING01 (TIMING01)	32	RW	6700_6A00h
38h	TSM_TIMING02 (TIMING02)	32	RW	6700_FFFFh
3Ch	TSM_TIMING03 (TIMING03)	32	RW	6700_6A00h
40h	TSM_TIMING04 (TIMING04)	32	RW	6700_6A00h
44h	TSM_TIMING05 (TIMING05)	32	RW	6700_6A00h
48h	TSM_TIMING06 (TIMING06)	32	RW	6700_6A00h
4Ch	TSM_TIMING07 (TIMING07)	32	RW	0500_0500h
50h	TSM_TIMING08 (TIMING08)	32	RW	0300_0300h
54h	TSM_TIMING09 (TIMING09)	32	RW	0300_0300h
58h	TSM_TIMING10 (TIMING10)	32	RW	6703_6A03h
5Ch	TSM_TIMING11 (TIMING11)	32	RW	FFFF_6A03h
60h	TSM_TIMING12 (TIMING12)	32	RW	6703_FFFFh
64h	TSM_TIMING13 (TIMING13)	32	RW	1600_4A00h
68h	TSM_TIMING14 (TIMING14)	32	RW	672F_645Fh
6Ch	TSM_TIMING15 (TIMING15)	32	RW	6703_6A03h
70h	TSM_TIMING16 (TIMING16)	32	RW	671A_FFFFh
74h	TSM_TIMING17 (TIMING17)	32	RW	FFFF_6A5Ah
78h	TSM_TIMING18 (TIMING18)	32	RW	6705_6A05h
7Ch	TSM_TIMING19 (TIMING19)	32	RW	1605_4A05h
80h	TSM_TIMING20 (TIMING20)	32	RW	6705_6A05h
84h	TSM_TIMING21 (TIMING21)	32	RW	6704_6A04h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
88h	TSM_TIMING22 (TIMING22)	32	RW	6704_FFFFh
8Ch	TSM_TIMING23 (TIMING23)	32	RW	FFFF_6A04h
90h	TSM_TIMING24 (TIMING24)	32	RW	1600_4A00h
94h	TSM_TIMING25 (TIMING25)	32	RW	671B_FFFFh
98h	TSM_TIMING26 (TIMING26)	32	RW	FFFF_6A5Ah
9Ch	TSM_TIMING27 (TIMING27)	32	RW	671E_FFFFh
A0h	TSM_TIMING28 (TIMING28)	32	RW	1F1E_FFFFh
A4h	TSM_TIMING29 (TIMING29)	32	RW	671C_FFFFh
A8h	TSM_TIMING30 (TIMING30)	32	RW	671E_FFFFh
ACh	TSM_TIMING31 (TIMING31)	32	RW	671D_FFFFh
B0h	TSM_TIMING32 (TIMING32)	32	RW	671B_FFFFh
B4h	TSM_TIMING33 (TIMING33)	32	RW	671E_FFFFh
B8h	TSM_TIMING34 (TIMING34)	32	RW	6705_6A05h
BCh	TSM_TIMING35 (TIMING35)	32	RW	FFFF_6A5Dh
C0h	TSM_TIMING36 (TIMING36)	32	RW	6764_FFFFh
C4h	TSM_TIMING37 (TIMING37)	32	RW	6564_FFFFh
C8h	TSM_TIMING38 (TIMING38)	32	RW	670C_6A40h
CCh	TSM_TIMING39 (TIMING39)	32	RW	6764_FFFFh
D0h	TSM_TIMING40 (TIMING40)	32	RW	6724_FFFFh
D4h	TSM_TIMING41 (TIMING41)	32	RW	2524_FFFFh
D8h	TSM_TIMING42 (TIMING42)	32	RW	FFFF_FFFFh
DCh	TSM_TIMING43 (TIMING43)	32	RW	FFFF_FFFFh
E0h	TSM_TIMING44 (TIMING44)	32	RW	FFFF_FFFFh
E4h	TSM_TIMING45 (TIMING45)	32	RW	FFFF_FFFFh
E8h	TSM_TIMING46 (TIMING46)	32	RW	FFFF_FFFFh
ECh	TSM_TIMING47 (TIMING47)	32	RW	FFFF_FFFFh
F0h	TSM_TIMING48 (TIMING48)	32	RW	FFFF_FFFFh
F4h	TSM_TIMING49 (TIMING49)	32	RW	FFFF_FFFFh
F8h	TSM_TIMING50 (TIMING50)	32	RW	FFFF_FFFFh
FCh	TSM_TIMING51 (TIMING51)	32	RW	6703_FFFFh
100h	TSM_TIMING52 (TIMING52)	32	RW	1504_FFFFh
104h	TSM_TIMING53 (TIMING53)	32	RW	6704_FFFFh
108h	TSM_TIMING54 (TIMING54)	32	RW	1504_FFFFh
10Ch	TSM_TIMING55 (TIMING55)	32	RW	671E_FFFFh
110h	TSM_TIMING56 (TIMING56)	32	RW	671E_FFFFh
114h	TSM_TIMING57 (TIMING57)	32	RW	1A03_FFFFh
118h	TSM_TIMING58 (TIMING58)	32	RW	FFFF_6A03h
11Ch	TSM OVERRIDE REGISTER 0 (OVRD0)	32	RW	0000_0000h
120h	TSM OVERRIDE REGISTER 1 (OVRD1)	32	RW	0000_0000h
124h	TSM OVERRIDE REGISTER 2 (OVRD2)	32	RW	0000_0000h

Table continues on the next page...

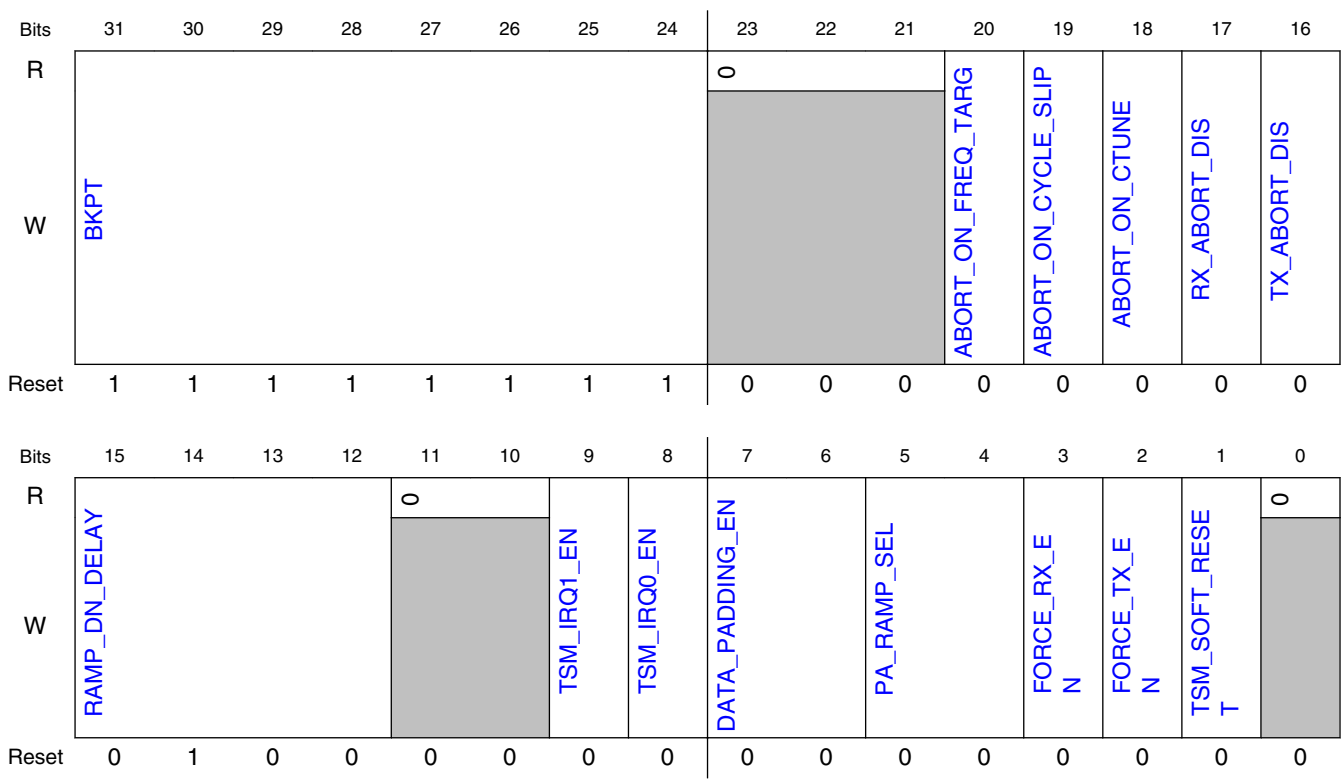
Offset	Register	Width (In bits)	Access	Reset value
128h	TSM OVERRIDE REGISTER 3 (OVRD3)	32	RW	0000_0000h

45.3.5.3.1.2 TSM CONTROL (CTRL)

45.3.5.3.1.2.1 Offset

Register	Offset
CTRL	0h

45.3.5.3.1.2.2 Diagram



45.3.5.3.1.2.3 Fields

Field	Function
31-24	TSM Breakpoint
BKPT	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

Table continues on the next page...

Field	Function
	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
20 ABORT_ON_FREQ_TARG	Abort On Frequency Target Lock Detect Failure 0b - don't allow TSM abort on Frequency Target Unlock Detect 1b - allow TSM abort on Frequency Target Unlock Detect
19 ABORT_ON_CYCLE_SLIP	Abort On Cycle Slip Lock Detect Failure 0b - don't allow TSM abort on Cycle Slip Unlock Detect 1b - allow TSM abort on Cycle Slip Unlock Detect
18 ABORT_ON_CTUNE	Abort On Coarse Tune Lock Detect Failure 0b - don't allow TSM abort on Coarse Tune Unlock Detect 1b - allow TSM abort on Coarse Tune Unlock Detect
17 RX_ABORT_DIS	Receive Abort Disable RX Abort disable. When set, prevents PLL unlock events during RX sequences from aborting the sequence.
16 TX_ABORT_DIS	Transmit Abort Disable TX Abort disable. When set, prevents PLL unlock events during TX sequences from aborting the sequence.
15-12 RAMP_DN_DELAY	PA Ramp Down Delay Delays the start of the PA Ramp Down, relative to the start of the TSM warmdown, by *N* microseconds, where *N*=RAMP_DN_DELAY. Range is 0 to 15us.
11-10 —	Reserved
9 TSM_IRQ1_EN	TSM_IRQ1 Enable/Disable bit TSM_IRQ1 Enable/Disable bit. Intended for debug purposes only. 0b - TSM_IRQ1 is disabled 1b - TSM_IRQ1 is enabled
8 TSM_IRQ0_EN	TSM_IRQ0 Enable/Disable bit TSM_IRQ0 Enable/Disable bit. Intended for debug purposes only. 0b - TSM_IRQ0 is disabled 1b - TSM_IRQ0 is enabled
7-6 DATA_PADDING_EN	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]. 00b - Disable TX Data Padding 01b - Enable TX Data Padding
5-4 PA_RAMP_SEL	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 Ramp Table. There are always 16 ramp steps, if ramping is enabled. The following table lists the total ramp duration for each PA_RAMP_SEL setting:

Table continues on the next page...

Field	Function
	0: No ramp 1: 1.0µs 2: 2.0µs 3: 4.0µs The following table lists the duration of each ramp step for each PA_RAMP_SEL setting: 0: No ramp 1: 2 Reference Clocks (0.0625µs) 2: 4 Reference Clocks (0.125µs) 3: 8 Reference Clocks (0.25µs)
3 FORCE_RX_EN	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. 0b - TSM Idle 1b - 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. 0b - TSM Idle 1b - TSM executes a TX sequence
1 TSM_SOFT_RESET	TSM Soft Reset Writing 1 to this bit returns TSM to its IDLE state, with TSM_COUNT=0. Writing 0 to this bit removes the forced-reset condition and resumes normal operation. The bit is not self-clearing. For contingency purposes only 0b - TSM Soft Reset removed. Normal operation. 1b - TSM Soft Reset engaged. TSM forced to IDLE, and holds there until the bit is cleared.
0 —	Reserved

45.3.5.3.1.3 TSM END OF SEQUENCE (END_OF_SEQ)

45.3.5.3.1.3.1 Offset

Register	Offset
END_OF_SEQ	4h

45.3.5.3.1.3.2 Diagram

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

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

45.3.5.3.1.3.3 Fields

Field	Function
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.
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.
7-0 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.

45.3.5.3.1.4 PA POWER (PA_POWER)

45.3.5.3.1.4.1 Offset

Register	Offset
PA_POWER	8h

45.3.5.3.1.4.2 Function

This contents of this register are used as PA target power when XCVR_CTRL[TGT_PWR_SRC] = 0

45.3.5.3.1.4.3 Diagram

Bits	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

Bits	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

45.3.5.3.1.4.4 Fields

Field	Function
31-6 —	Reserved
5-0 PA_POWER	PA POWER This contents of this register are used as PA target power when XCVR_CTRL[TGT_PWR_SRC] = 0.The valid values for this field are 0,1,2,4,6,8,…,62 (in increasing order of PA power). That is, above 1, only even numbers are permitted in the PA_POWER bitfield. Odd values of 3..63 are not permitted and will result in unexpected behavior.

45.3.5.3.1.5 PA RAMP TABLE 0 (PA_RAMP_TBL0)

45.3.5.3.1.5.1 Offset

Register	Offset
PA_RAMP_TBL0	Ch

45.3.5.3.1.5.2 Function

PA Ramp Table 0

45.3.5.3.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		PA_RAMP3						0		PA_RAMP2					
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PA_RAMP1						0		PA_RAMP0					
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

45.3.5.3.1.5.4 Fields

Field	Function
31-30 —	Reserved
29-24 PA_RAMP3	PA_RAMP3 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 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_RAMP3 cannot exceed target power (enforced by PA ramping logic).
23-22 —	Reserved
21-16 PA_RAMP2	PA_RAMP2 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 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_RAMP2 cannot exceed target power (enforced by PA ramping logic).
15-14 —	Reserved
13-8 PA_RAMP1	PA_RAMP1 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_RAMP1 cannot exceed target power (enforced by PA ramping logic).
7-6 —	Reserved
5-0 PA_RAMP0	PA_RAMP0 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_pa_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_RAMP0 cannot exceed target power (enforced by PA ramping logic). When PA ramping is enabled, the contents of PA_RAMP0 are also presented to the PA during sequence-idle conditions.

45.3.5.3.1.6 PA RAMP TABLE 1 (PA_RAMP_TBL1)

45.3.5.3.1.6.1 Offset

Register	Offset
PA_RAMP_TBL1	10h

45.3.5.3.1.6.2 Function

PA Ramp Table 1

45.3.5.3.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		PA_RAMP7						0		PA_RAMP6					
W																
Reset	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PA_RAMP5						0		PA_RAMP4					
W																
Reset	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0

45.3.5.3.1.6.4 Fields

Field	Function
31-30 —	Reserved
29-24 PA_RAMP7	PA_RAMP7 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 eighth ramp step. During PA ramp down, the contents of this register are the PA power value during the eighth-to-last ramp step. In both cases, PA_RAMP7 cannot exceed target power (enforced by PA ramping logic).
23-22 —	Reserved
21-16 PA_RAMP6	PA_RAMP6 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 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_RAMP6 cannot exceed target power (enforced by PA ramping logic).
15-14	Reserved

Table continues on the next page...

Field	Function
—	
13-8 PA_RAMP5	PA_RAMP5 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 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_RAMP5 cannot exceed target power (enforced by PA ramping logic).
7-6 —	Reserved
5-0 PA_RAMP4	PA_RAMP4 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 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_RAMP4 cannot exceed target power (enforced by PA ramping logic).

45.3.5.3.1.7 PA RAMP TABLE 2 (PA_RAMP_TBL2)

45.3.5.3.1.7.1 Offset

Register	Offset
PA_RAMP_TBL2	14h

45.3.5.3.1.7.2 Function

PA Ramp Table 2

45.3.5.3.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		PA_RAMP11						0		PA_RAMP10					
W																
Reset	0	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PA_RAMP9						0		PA_RAMP8					
W																
Reset	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0

45.3.5.3.1.7.4 Fields

Field	Function
31-30 —	Reserved
29-24 PA_RAMP11	PA_RAMP11 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 twelvth ramp step. During PA ramp down, the contents of this register are the PA power value during the twelvth-to-last ramp step. In both cases, PA_RAMP11 cannot exceed target power (enforced by PA ramping logic).
23-22 —	Reserved
21-16 PA_RAMP10	PA_RAMP10 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 eleventh ramp step. During PA ramp down, the contents of this register are the PA power value during the eleventh-to-last ramp step. In both cases, PA_RAMP10 cannot exceed target power (enforced by PA ramping logic).
15-14 —	Reserved
13-8 PA_RAMP9	PA_RAMP9 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 tenth ramp step. During PA ramp down, the contents of this register are the PA power value during the tenth-to-last ramp step. In both cases, PA_RAMP9 cannot exceed target power (enforced by PA ramping logic).
7-6 —	Reserved
5-0 PA_RAMP8	PA_RAMP8 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 ninth ramp step. During PA ramp down, the contents of this register are the PA power value during the ninth-to-last ramp step. In both cases, PA_RAMP8 cannot exceed target power (enforced by PA ramping logic).

45.3.5.3.1.8 PA RAMP TABLE 3 (PA_RAMP_TBL3)

45.3.5.3.1.8.1 Offset

Register	Offset
PA_RAMP_TBL3	18h

45.3.5.3.1.8.2 Function

PA Ramp Table 3

45.3.5.3.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		PA_RAMP15						0		PA_RAMP14					
W																
Reset	0	0	1	1	1	1	0	0	0	0	1	1	0	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PA_RAMP13						0		PA_RAMP12					
W																
Reset	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0	0

45.3.5.3.1.8.4 Fields

Field	Function
31-30 —	Reserved
29-24 PA_RAMP15	PA_RAMP15 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 sixteenth (final) ramp step. During PA ramp down, the contents of this register are the PA power value during the sixteenth-to-last (first) ramp step. In both cases, PA_RAMP15 cannot exceed target power (enforced by PA ramping logic).
23-22 —	Reserved
21-16 PA_RAMP14	PA_RAMP14 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 fifteenth ramp step. During PA ramp down, the contents of this register are the PA power value during the fifteenth-to-last ramp step. In both cases, PA_RAMP14 cannot exceed target power (enforced by PA ramping logic).
15-14 —	Reserved
13-8 PA_RAMP13	PA_RAMP13 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 fourteenth ramp step. During PA ramp down, the contents of this register are the PA power value during the fourteenth-to-last ramp step. In both cases, PA_RAMP13 cannot exceed target power (enforced by PA ramping logic).
7-6 —	Reserved
5-0 PA_RAMP12	PA_RAMP12 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 thirteenth ramp step. During PA ramp down, the contents of this register are the PA power value during the thirteenth-to-last ramp step. In both cases, PA_RAMP12 cannot exceed target power (enforced by PA ramping logic).

45.3.5.3.1.9 TSM RECYCLE COUNT (RECYCLE_COUNT)

45.3.5.3.1.9.1 Offset

Register	Offset
RECYCLE_COUNT	24h

45.3.5.3.1.9.2 Function

This register contains the TSM Recycle "Jump-to" points for the 3 types of TSM Recycle

45.3.5.3.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								RECYCLE_COUNT2							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RECYCLE_COUNT1								RECYCLE_COUNT0							
W																
Reset	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0

45.3.5.3.1.9.4 Fields

Field	Function
31-24 —	Reserved
23-16 RECYCLE_COUNT2	TSM RX Recycle Count 2 Reserved
15-8 RECYCLE_COUNT1	TSM RX Recycle Count 1 Reserved
7-0 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 Generic_FSK Link Layer asserts a tsm_recycle to TSM. 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 any protocol engine to TSM, requesting a 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".

45.3.5.3.1.10 TSM FAST WARMUP CONTROL 1 (FAST_CTRL1)

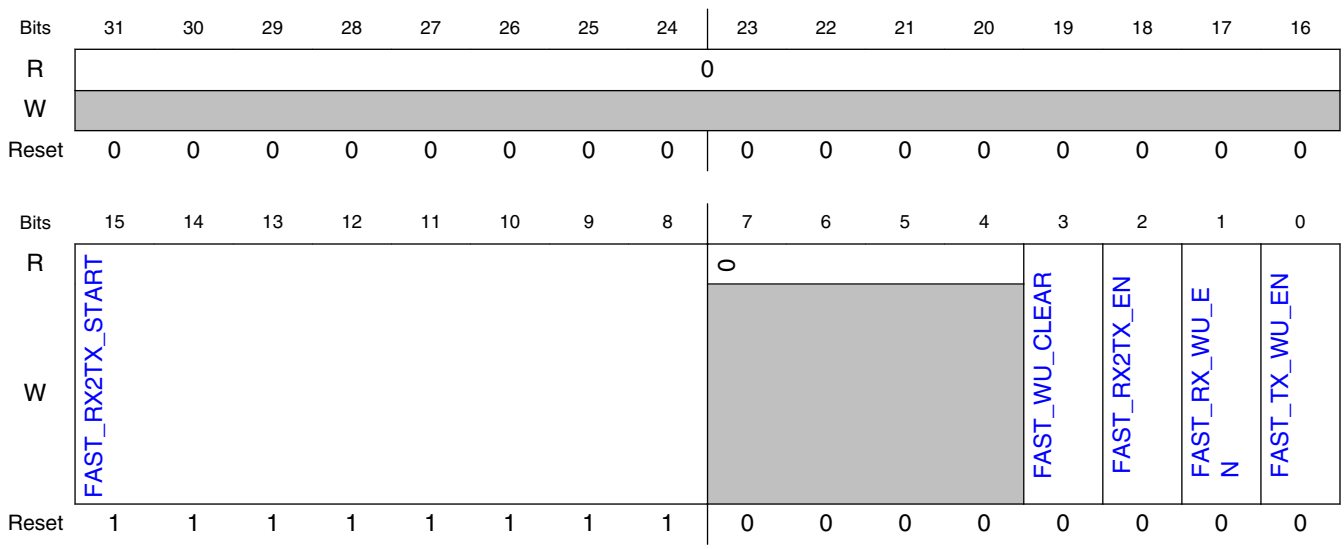
45.3.5.3.1.10.1 Offset

Register	Offset
FAST_CTRL1	28h

45.3.5.3.1.10.2 Function

This register provides enabling and control for Fast TSM Mode

45.3.5.3.1.10.3 Diagram



45.3.5.3.1.10.4 Fields

Field	Function
31-16 —	Reserved
15-8 FAST_RX2TX_START	These bits currently have no functionality.
7-4 —	Reserved
3	Fast TSM Warmup Clear State

Table continues on the next page...

Field	Function
FAST_WU_CLEAR	This bit clears the RF channel memory in the PLL Digital Block, forcing the next TSM TX Warmup to be normal (not fast), and the next TSM RX Warmup to be normal (not fast), regardless of whether the RF channel has actually changed or not. This bit is not self-clearing. Write '1' to clear channel memory, then write '0' to proceed with TSM operations.
2 FAST_RX2TX_EN	Fast TSM RX-to-TX Transition Enable This bit currently has no functionality.
1 FAST_RX_WU_EN	Fast TSM RX Warmup Enable 0b - Fast TSM RX Warmups are disabled 1b - Fast TSM RX Warmups are enabled, if the RF channel has not changed since the last RX warmup, and for BLE mode, the RF channel is not an advertising channel.
0 FAST_TX_WU_EN	Fast TSM TX Warmup Enable 0b - Fast TSM TX Warmups are disabled 1b - Fast TSM TX Warmups are enabled, if the RF channel has not changed since the last TX warmup, and for BLE mode, the RF channel is not an advertising channel.

45.3.5.3.1.11 TSM FAST WARMUP CONTROL 2 (FAST_CTRL2)

45.3.5.3.1.11.1 Offset

Register	Offset
FAST_CTRL2	2Ch

45.3.5.3.1.11.2 Function

This register provides configuration for Fast TSM Mode

45.3.5.3.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FAST_DEST_RX								FAST_START_RX							
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	FAST_DEST_TX								FAST_START_TX							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

45.3.5.3.1.11.4 Fields

Field	Function
31-24 FAST_DEST_RX	Fast TSM RX "Jump-to" Point During a Fast RX Warmup, this value represents the ending TSM count for the portion of the warmup that is to be skipped over. If Fast RX Warmups are enabled, FAST_START_RX[7:0] will never be reached and will instead be replaced by the contents of this register, thereby executing the jump. Example: FAST_START_RX = 10 FAST_DEST_RX = 15 The TSM will count as follows: … 7,8,9,15,16,17 …
23-16 FAST_START_RX	Fast TSM RX "Jump-from" Point During a Fast RX Warmup, this value represents the starting TSM count for the portion of the warmup that is to be skipped over. If Fast RX Warmups are enabled, this count will never be reached; its place in the TSM warmup will be replaced by the contents of FAST_DEST_RX[7:0], thereby executing the jump.
15-8 FAST_DEST_TX	Fast TSM TX "Jump-to" Point During a Fast TX Warmup, this value represents the ending TSM count for the portion of the warmup that is to be skipped over. If Fast TX Warmups are enabled, FAST_START_TX[7:0] will never be reached and will instead be replaced by the contents of this register, thereby executing the jump. Example: FAST_START_TX = 10 FAST_DEST_TX = 15 The TSM will count as follows: … 7,8,9,15,16,17 …
7-0 FAST_START_TX	Fast TSM TX "Jump-from" Point During a Fast TX Warmup, this value represents the starting TSM count for the portion of the warmup that is to be skipped over. If Fast TX Warmups are enabled, this count will never be reached; its place in the TSM warmup will be replaced by the contents of FAST_DEST_TX[7:0], thereby executing the jump.

45.3.5.3.1.12 TSM_TIMING00 (TIMING00)

45.3.5.3.1.12.1 Offset

Register	Offset
TIMING00	30h

45.3.5.3.1.12.2 Function

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the BB_LDO_HF_EN TSM signal or signal group.

45.3.5.3.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_HF_EN_RX_LO								BB_LDO_HF_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_HF_EN_TX_LO								BB_LDO_HF_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

45.3.5.3.1.12.4 Fields

Field	Function
31-24 BB_LDO_HF_EN_RX_LO	De-assertion time setting for BB_LDO_HF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_HF_EN signal or group will transition from HI to LO.
23-16 BB_LDO_HF_EN_RX_HI	Assertion time setting for BB_LDO_HF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_HF_EN signal or group will transition from LO to HI.
15-8 BB_LDO_HF_EN_TX_LO	De-assertion time setting for BB_LDO_HF_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_HF_EN signal or group will transition from HI to LO.
7-0 BB_LDO_HF_EN_TX_HI	Assertion time setting for BB_LDO_HF_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_HF_EN signal or group will transition from LO to HI.

45.3.5.3.1.13 TSM_TIMING01 (TIMING01)

45.3.5.3.1.13.1 Offset

Register	Offset
TIMING01	34h

45.3.5.3.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_ADCDAC_EN_RX_LO								BB_LDO_ADCDAC_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_ADCDAC_EN_TX_LO								BB_LDO_ADCDAC_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

45.3.5.3.1.13.3 Fields

Field	Function
31-24 BB_LDO_ADCDAC_EN_RX_LO	De-assertion time setting for BB_LDO_ADCDAC_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_ADCDAC_EN signal or group will transition from HI to LO.
23-16 BB_LDO_ADCDAC_EN_RX_HI	Assertion time setting for BB_LDO_ADCDAC_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_ADCDAC_EN signal or group will transition from LO to HI.
15-8 BB_LDO_ADCDAC_EN_TX_LO	De-assertion time setting for BB_LDO_ADCDAC_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_ADCDAC_EN signal or group will transition from HI to LO.
7-0 BB_LDO_ADCDAC_EN_TX_HI	Assertion time setting for BB_LDO_ADCDAC_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_ADCDAC_EN signal or group will transition from LO to HI.

45.3.5.3.1.14 TSM_TIMING02 (TIMING02)

45.3.5.3.1.14.1 Offset

Register	Offset
TIMING02	38h

45.3.5.3.1.14.2 Diagram

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

45.3.5.3.1.14.3 Fields

Field	Function
31-24 BB_LDO_BBA_EN_RX_LO	De-assertion time setting for BB_LDO_BBA_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_BBA_EN signal or group will transition from HI to LO.
23-16 BB_LDO_BBA_EN_RX_HI	Assertion time setting for BB_LDO_BBA_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_BBA_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.15 TSM_TIMING03 (TIMING03)

45.3.5.3.1.15.1 Offset

Register	Offset
TIMING03	3Ch

45.3.5.3.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_PD_EN_RX_LO								BB_LDO_PD_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_PD_EN_TX_LO								BB_LDO_PD_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

45.3.5.3.1.15.3 Fields

Field	Function
31-24 BB_LDO_PD_EN_RX_LO	De-assertion time setting for BB_LDO_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_PD_EN signal or group will transition from HI to LO.
23-16 BB_LDO_PD_EN_RX_HI	Assertion time setting for BB_LDO_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_PD_EN signal or group will transition from LO to HI.
15-8 BB_LDO_PD_EN_TX_LO	De-assertion time setting for BB_LDO_PD_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_PD_EN signal or group will transition from HI to LO.
7-0 BB_LDO_PD_EN_TX_HI	Assertion time setting for BB_LDO_PD_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_PD_EN signal or group will transition from LO to HI.

45.3.5.3.1.16 TSM_TIMING04 (TIMING04)

45.3.5.3.1.16.1 Offset

Register	Offset
TIMING04	40h

45.3.5.3.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_FDBK_EN_RX_LO								BB_LDO_FDBK_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_FDBK_EN_TX_LO								BB_LDO_FDBK_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

45.3.5.3.1.16.3 Fields

Field	Function
31-24 BB_LDO_FDBK_EN_RX_LO	De-assertion time setting for BB_LDO_FDBK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_EN signal or group will transition from HI to LO.
23-16 BB_LDO_FDBK_EN_RX_HI	Assertion time setting for BB_LDO_FDBK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_EN signal or group will transition from LO to HI.
15-8 BB_LDO_FDBK_EN_TX_LO	De-assertion time setting for BB_LDO_FDBK_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_EN signal or group will transition from HI to LO.
7-0 BB_LDO_FDBK_EN_TX_HI	Assertion time setting for BB_LDO_FDBK_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_EN signal or group will transition from LO to HI.

45.3.5.3.1.17 TSM_TIMING05 (TIMING05)

45.3.5.3.1.17.1 Offset

Register	Offset
TIMING05	44h

45.3.5.3.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_VCOLO_EN_RX_LO								BB_LDO_VCOLO_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_VCOLO_EN_TX_LO								BB_LDO_VCOLO_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

45.3.5.3.1.17.3 Fields

Field	Function
31-24 BB_LDO_VCOLO_EN_RX_LO	De-assertion time setting for BB_LDO_VCOLO_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_EN signal or group will transition from HI to LO.
23-16 BB_LDO_VCOLO_EN_RX_HI	Assertion time setting for BB_LDO_VCOLO_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_EN signal or group will transition from LO to HI.
15-8 BB_LDO_VCOLO_EN_TX_LO	De-assertion time setting for BB_LDO_VCOLO_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_EN signal or group will transition from HI to LO.
7-0 BB_LDO_VCOLO_EN_TX_HI	Assertion time setting for BB_LDO_VCOLO_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_EN signal or group will transition from LO to HI.

45.3.5.3.1.18 TSM_TIMING06 (TIMING06)

45.3.5.3.1.18.1 Offset

Register	Offset
TIMING06	48h

45.3.5.3.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_VTREF_EN_RX_LO								BB_LDO_VTREF_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_VTREF_EN_TX_LO								BB_LDO_VTREF_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

45.3.5.3.1.18.3 Fields

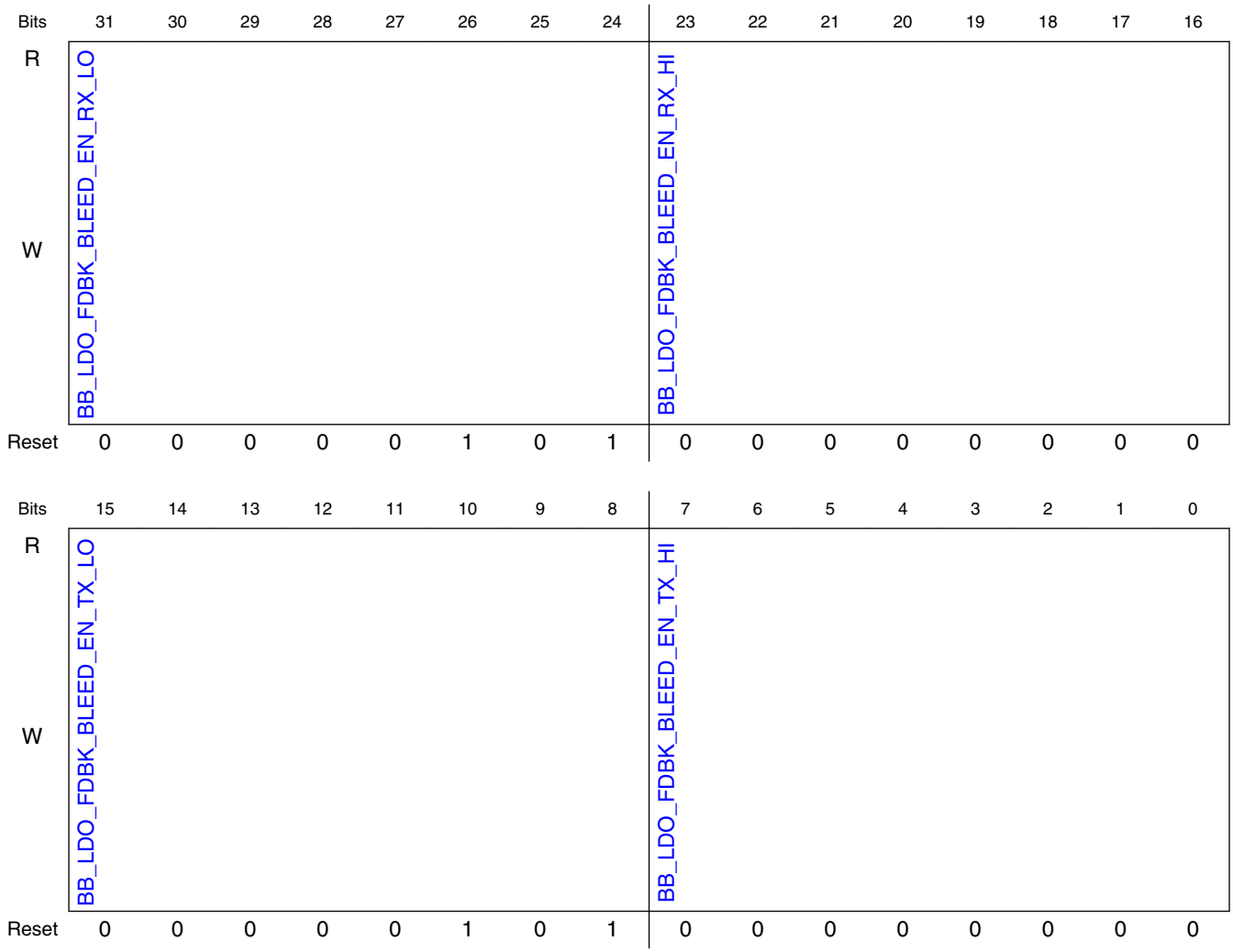
Field	Function
31-24 BB_LDO_VTREF_EN_RX_LO	De-assertion time setting for BB_LDO_VTREF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VTREF_EN signal or group will transition from HI to LO.
23-16 BB_LDO_VTREF_EN_RX_HI	Assertion time setting for BB_LDO_VTREF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VTREF_EN signal or group will transition from LO to HI.
15-8 BB_LDO_VTREF_EN_TX_LO	De-assertion time setting for BB_LDO_VTREF_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VTREF_EN signal or group will transition from HI to LO.
7-0 BB_LDO_VTREF_EN_TX_HI	Assertion time setting for BB_LDO_VTREF_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VTREF_EN signal or group will transition from LO to HI.

45.3.5.3.1.19 TSM_TIMING07 (TIMING07)

45.3.5.3.1.19.1 Offset

Register	Offset
TIMING07	4Ch

45.3.5.3.1.19.2 Diagram



45.3.5.3.1.19.3 Fields

Field	Function
31-24 BB_LDO_FDBK_BLEED_EN_RX_LO	De-assertion time setting for BB_LDO_FDBK_BLEED_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_BLEED_EN signal or group will transition from HI to LO.
23-16 BB_LDO_FDBK_BLEED_EN_RX_HI	Assertion time setting for BB_LDO_FDBK_BLEED_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_BLEED_EN signal or group will transition from LO to HI.
15-8 BB_LDO_FDBK_BLEED_EN_TX_LO	De-assertion time setting for BB_LDO_FDBK_BLEED_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_BLEED_EN signal or group will transition from HI to LO.

Table continues on the next page...

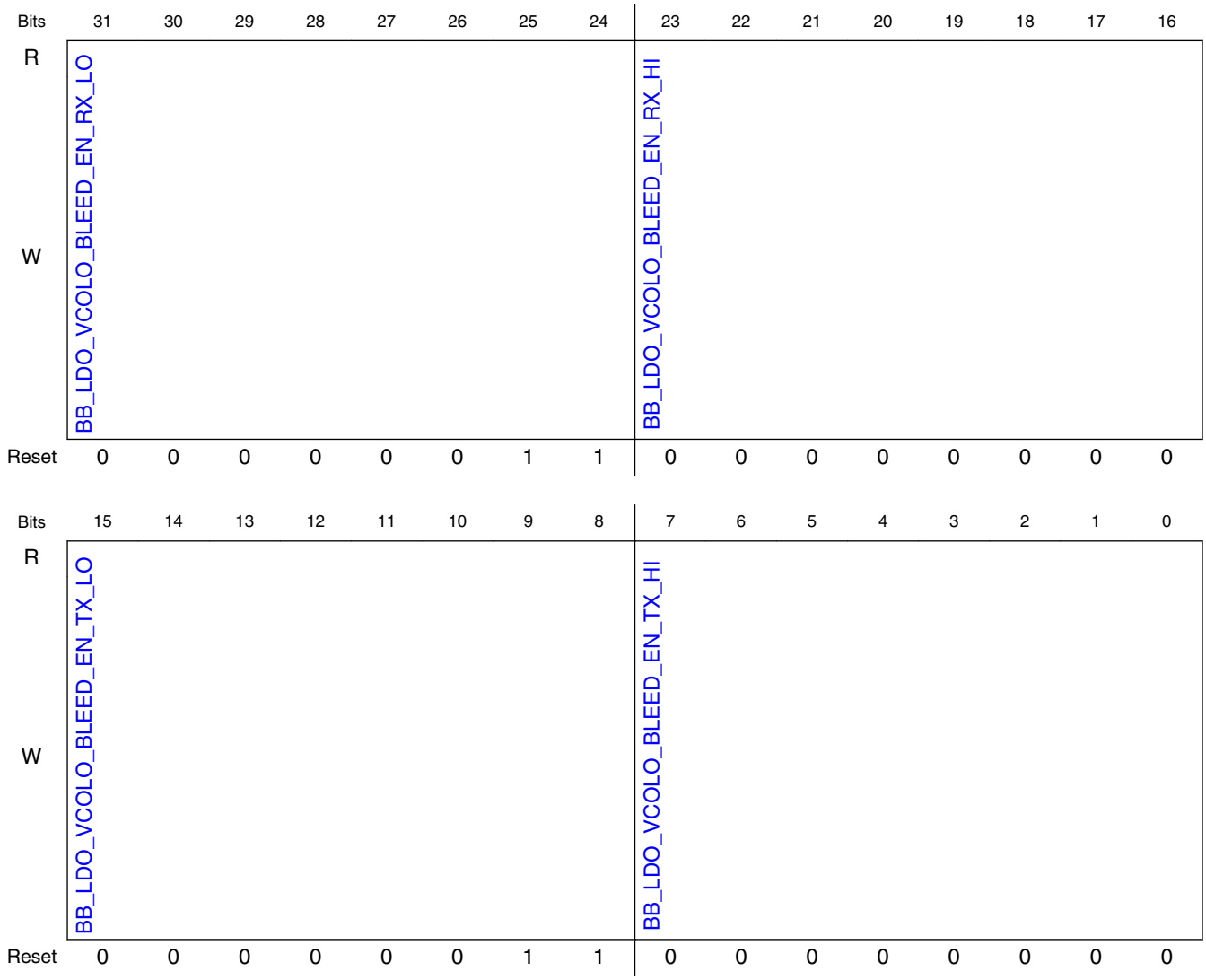
Field	Function
7-0 BB_LDO_FDBK_BLEED_EN_TX_HI	Assertion time setting for BB_LDO_FDBK_BLEED_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_BLEED_EN signal or group will transition from LO to HI.

45.3.5.3.1.20 TSM_TIMING08 (TIMING08)

45.3.5.3.1.20.1 Offset

Register	Offset
TIMING08	50h

45.3.5.3.1.20.2 Diagram



45.3.5.3.1.20.3 Fields

Field	Function
31-24 BB_LDO_VCOLO_BLEED_EN_RX_LO	De-assertion time setting for BB_LDO_VCOLO_BLEED_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_BLEED_EN signal or group will transition from HI to LO.
23-16 BB_LDO_VCOLO_BLEED_EN_RX_HI	Assertion time setting for BB_LDO_VCOLO_BLEED_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_BLEED_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for BB_LDO_VCOLO_BLEED_EN (TX)

Table continues on the next page...

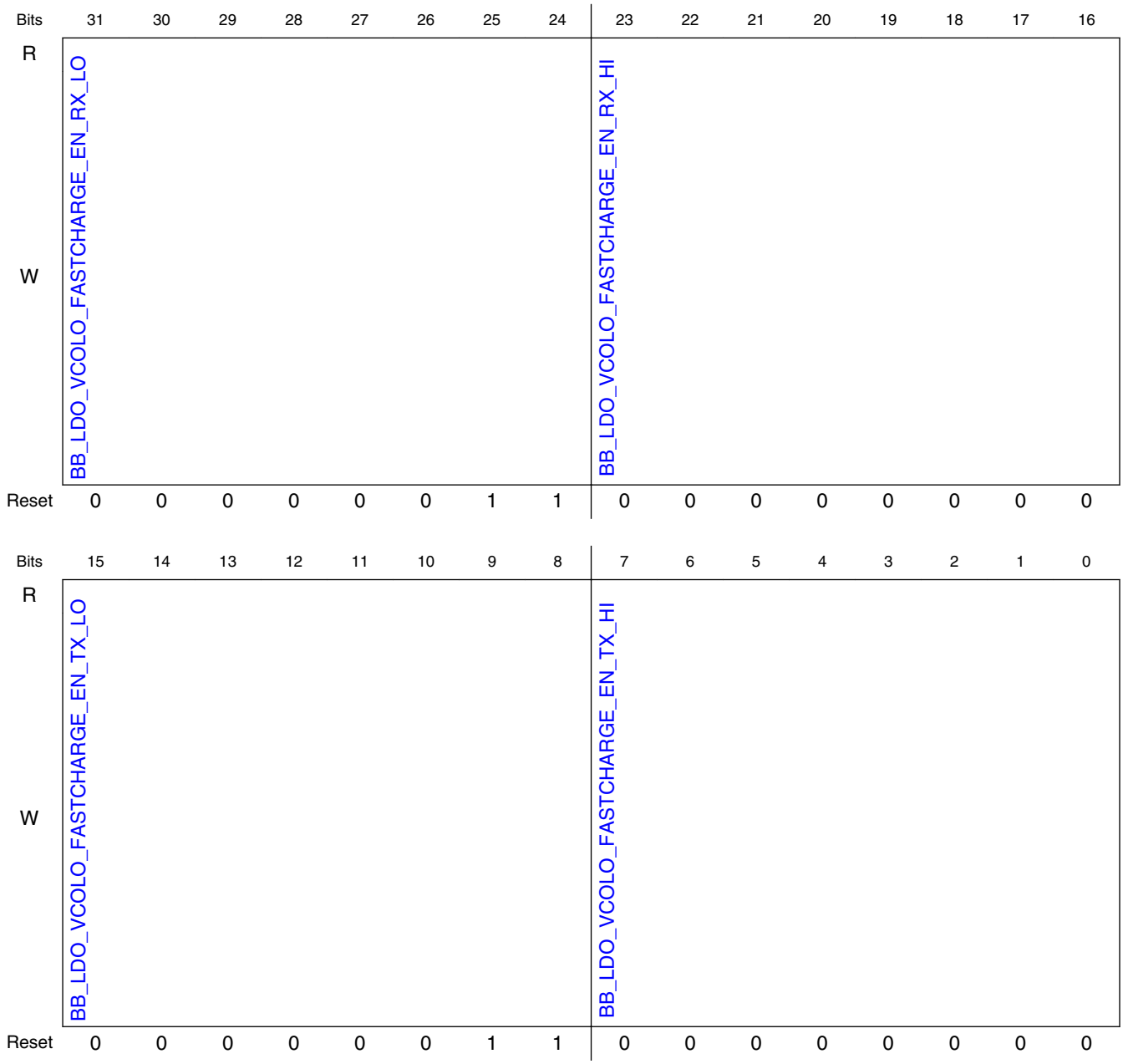
Field	Function
BB_LDO_VCOLO_BLEED_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_BLEED_EN signal or group will transition from HI to LO.
7-0 BB_LDO_VCOLO_BLEED_EN_TX_HI	Assertion time setting for BB_LDO_VCOLO_BLEED_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_BLEED_EN signal or group will transition from LO to HI.

45.3.5.3.1.21 TSM_TIMING09 (TIMING09)

45.3.5.3.1.21.1 Offset

Register	Offset
TIMING09	54h

45.3.5.3.1.21.2 Diagram



45.3.5.3.1.21.3 Fields

Field	Function
31-24 BB_LDO_VCOLO_FASTCHARGE_EN_RX_LO	De-assertion time setting for BB_LDO_VCOLO_FASTCHARGE_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_FASTCHARGE_EN signal or group will transition from HI to LO.
23-16	Assertion time setting for BB_LDO_VCOLO_FASTCHARGE_EN (RX)

Table continues on the next page...

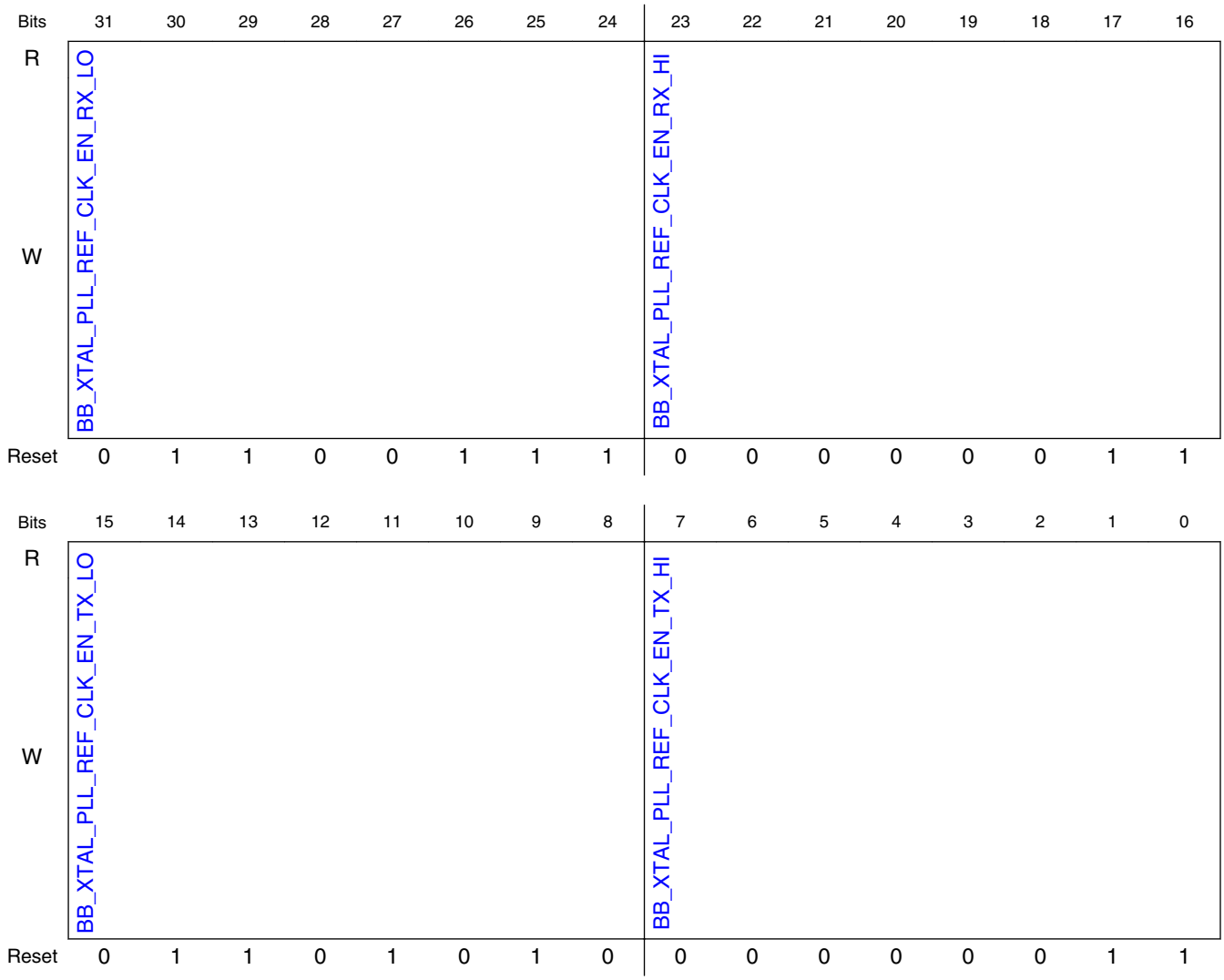
Field	Function
BB_LDO_VCOLO_FASTCHARGE_EN_RX_HI	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_FASTCHARGE_EN signal or group will transition from LO to HI.
15-8 BB_LDO_VCOLO_FASTCHARGE_EN_TX_LO	De-assertion time setting for BB_LDO_VCOLO_FASTCHARGE_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_FASTCHARGE_EN signal or group will transition from HI to LO.
7-0 BB_LDO_VCOLO_FASTCHARGE_EN_TX_HI	Assertion time setting for BB_LDO_VCOLO_FASTCHARGE_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_FASTCHARGE_EN signal or group will transition from LO to HI.

45.3.5.3.1.22 TSM_TIMING10 (TIMING10)

45.3.5.3.1.22.1 Offset

Register	Offset
TIMING10	58h

45.3.5.3.1.22.2 Diagram



45.3.5.3.1.22.3 Fields

Field	Function
31-24 BB_XTAL_PLL_REF_CLK_EN_RX_LO	De-assertion time setting for BB_XTAL_PLL_REF_CLK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_XTAL_PLL_REF_CLK_EN signal or group will transition from HI to LO.
23-16 BB_XTAL_PLL_REF_CLK_EN_RX_HI	Assertion time setting for BB_XTAL_PLL_REF_CLK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_XTAL_PLL_REF_CLK_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for BB_XTAL_PLL_REF_CLK_EN (TX)

Table continues on the next page...

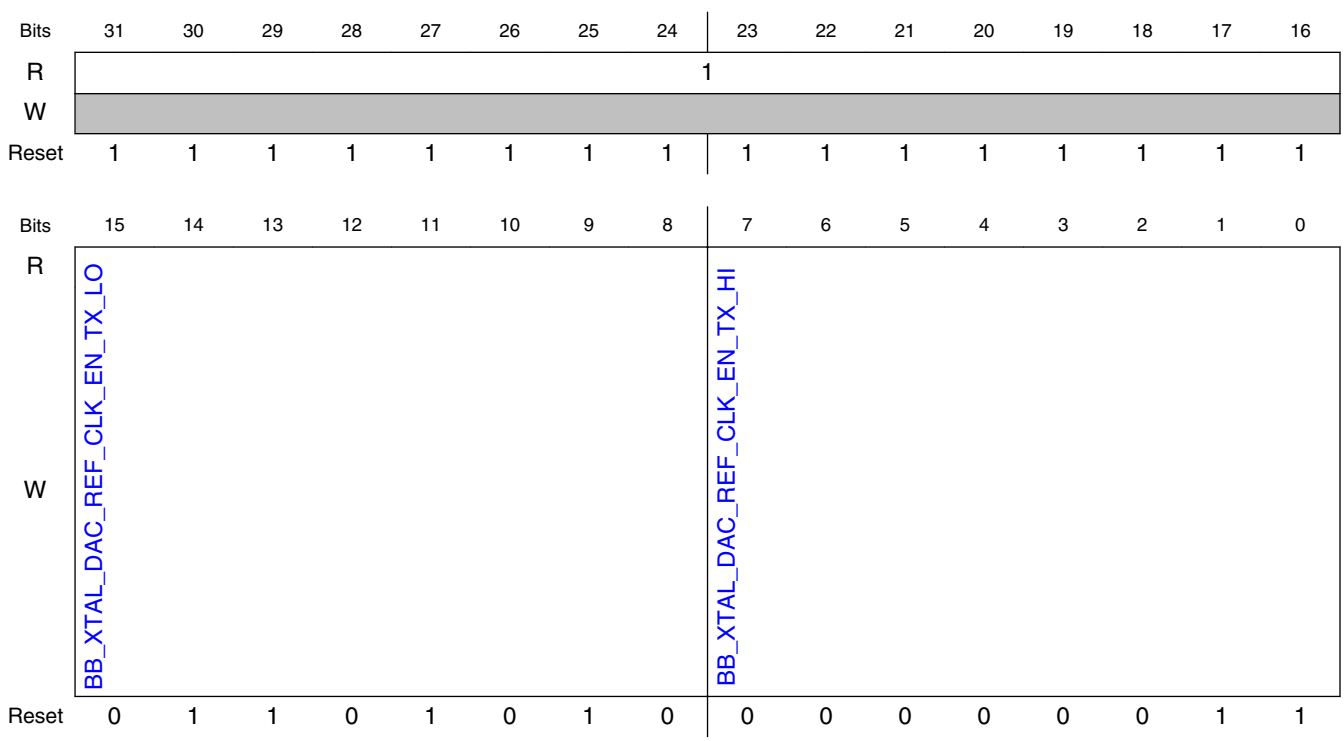
Field	Function
BB_XTAL_PLL_REF_CLK_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_XTAL_PLL_REF_CLK_EN signal or group will transition from HI to LO.
7-0 BB_XTAL_PLL_REF_CLK_EN_TX_HI	Assertion time setting for BB_XTAL_PLL_REF_CLK_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_XTAL_PLL_REF_CLK_EN signal or group will transition from LO to HI.

45.3.5.3.1.23 TSM_TIMING11 (TIMING11)

45.3.5.3.1.23.1 Offset

Register	Offset
TIMING11	5Ch

45.3.5.3.1.23.2 Diagram



45.3.5.3.1.23.3 Fields

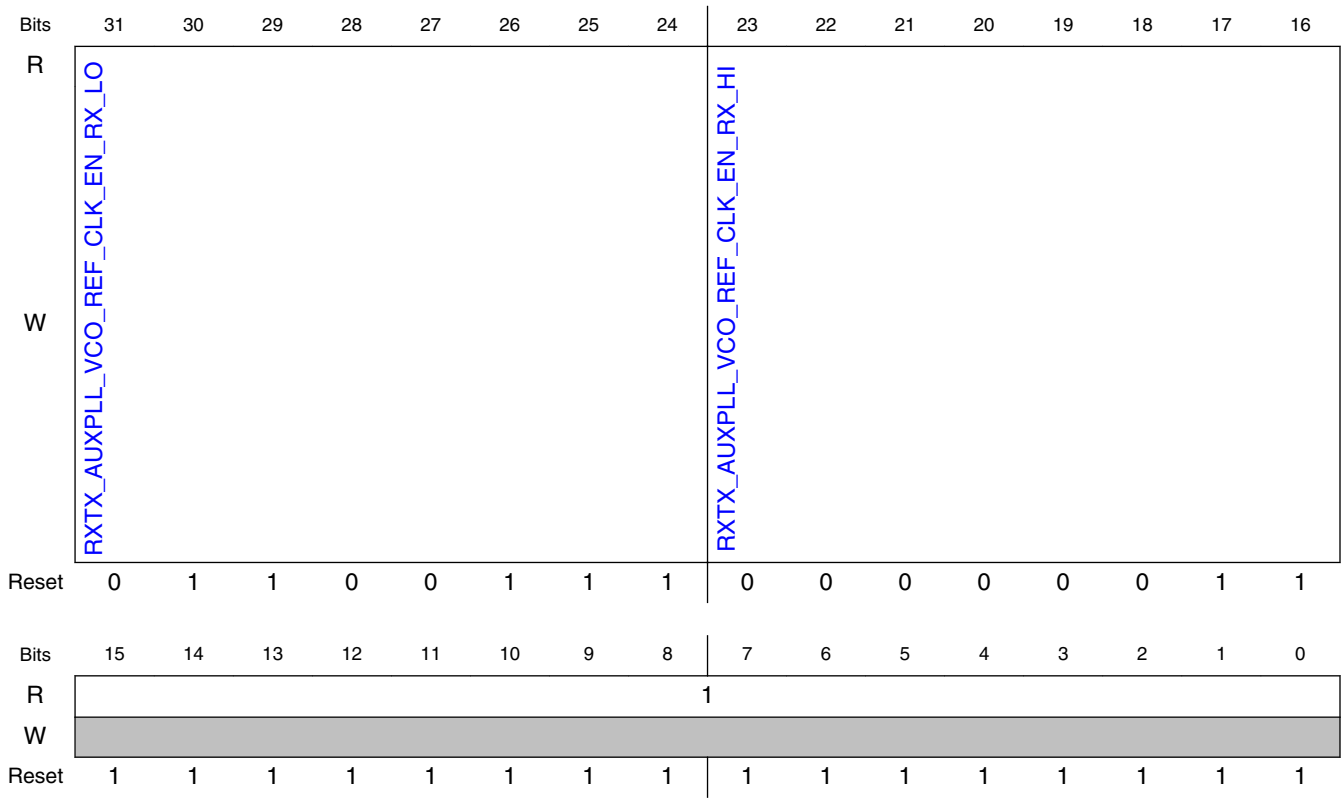
Field	Function
31-16 —	Reserved
15-8 BB_XTAL_DAC_REF_CLK_EN_TX_LO	De-assertion time setting for BB_XTAL_DAC_REF_CLK_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_XTAL_DAC_REF_CLK_EN signal or group will transition from HI to LO.
7-0 BB_XTAL_DAC_REF_CLK_EN_TX_HI	Assertion time setting for BB_XTAL_DAC_REF_CLK_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_XTAL_DAC_REF_CLK_EN signal or group will transition from LO to HI.

45.3.5.3.1.24 TSM_TIMING12 (TIMING12)

45.3.5.3.1.24.1 Offset

Register	Offset
TIMING12	60h

45.3.5.3.1.24.2 Diagram



45.3.5.3.1.24.3 Fields

Field	Function
31-24 RXTX_AUXPLL_VCO_REF_CLK_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_VCO_REF_CLK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_VCO_REF_CLK_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_VCO_REF_CLK_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_VCO_REF_CLK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_VCO_REF_CLK_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.25 TSM_TIMING13 (TIMING13)

45.3.5.3.1.25.1 Offset

Register	Offset
TIMING13	64h

45.3.5.3.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PLL_LOOP_IS_OPEN_RX_LO								PLL_LOOP_IS_OPEN_RX_HI							
Reset	0	0	0	1	0	1	1	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	PLL_LOOP_IS_OPEN_TX_LO								PLL_LOOP_IS_OPEN_TX_HI							
Reset	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

45.3.5.3.1.25.3 Fields

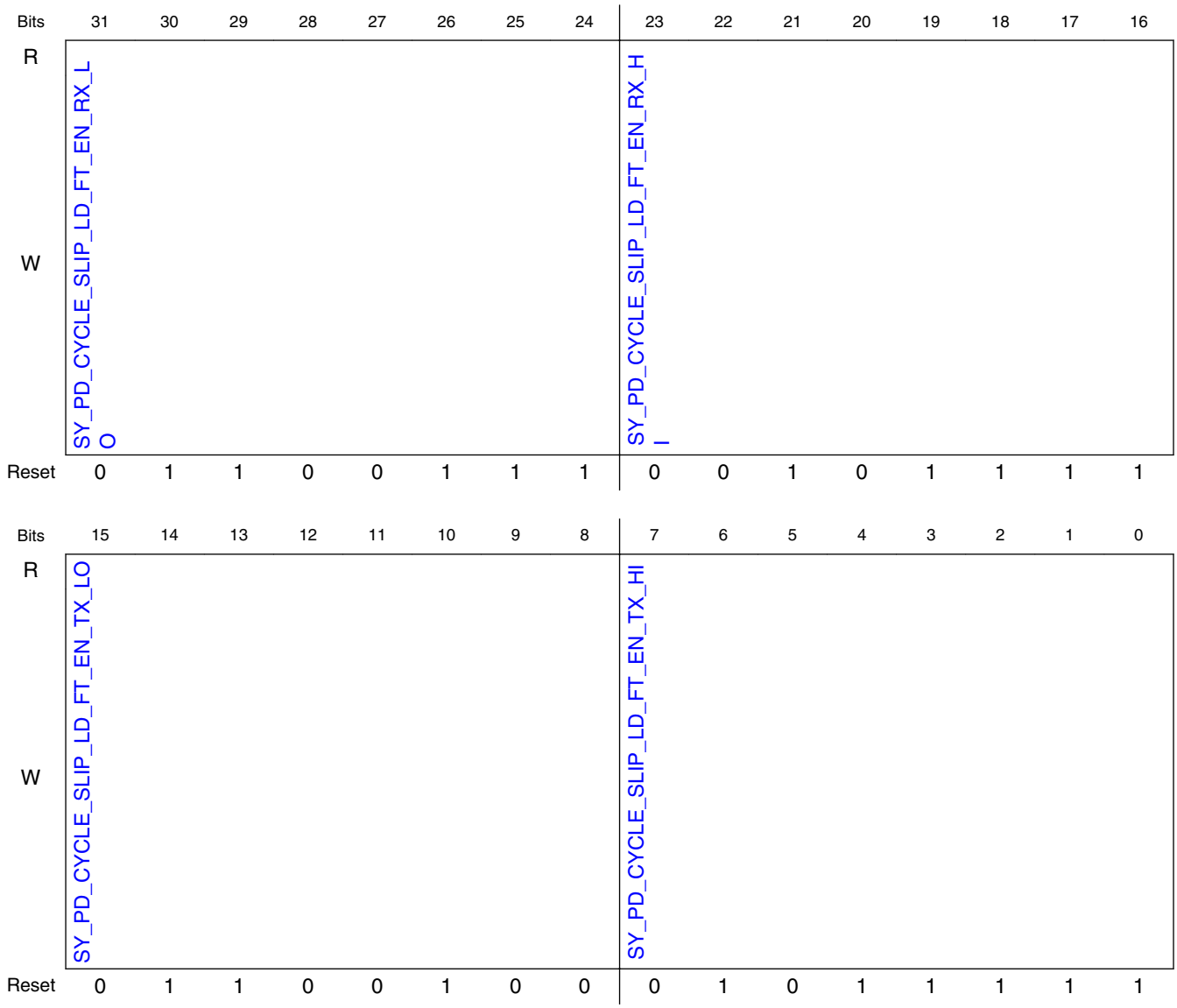
Field	Function
31-24 PLL_LOOP_IS_OPEN_RX_LO	De-assertion time setting for PLL_LOOP_IS_OPEN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_LOOP_IS_OPEN signal or group will transition from HI to LO.
23-16 PLL_LOOP_IS_OPEN_RX_HI	Assertion time setting for PLL_LOOP_IS_OPEN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_LOOP_IS_OPEN signal or group will transition from LO to HI.
15-8 PLL_LOOP_IS_OPEN_TX_LO	De-assertion time setting for PLL_LOOP_IS_OPEN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_LOOP_IS_OPEN signal or group will transition from HI to LO.
7-0 PLL_LOOP_IS_OPEN_TX_HI	Assertion time setting for PLL_LOOP_IS_OPEN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_LOOP_IS_OPEN signal or group will transition from LO to HI.

45.3.5.3.1.26 TSM_TIMING14 (TIMING14)

45.3.5.3.1.26.1 Offset

Register	Offset
TIMING14	68h

45.3.5.3.1.26.2 Diagram



45.3.5.3.1.26.3 Fields

Field	Function
31-24 SY_PD_CYCLE_SLIP_LD_FT_EN_RX_LO	De-assertion time setting for SY_PD_CYCLE_SLIP_LD_FT_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_CYCLE_SLIP_LD_FT_EN signal or group will transition from HI to LO.
23-16	Assertion time setting for SY_PD_CYCLE_SLIP_LD_FT_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_CYCLE_SLIP_LD_FT_EN signal or group will transition from LO to HI.

Table continues on the next page...

Field	Function
SY_PD_CYCLE_SLIP_LD_FT_EN_RX_HI	
15-8 SY_PD_CYCLE_SLIP_LD_FT_EN_TX_LO	De-assertion time setting for SY_PD_CYCLE_SLIP_LD_FT_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_CYCLE_SLIP_LD_FT_EN signal or group will transition from HI to LO.
7-0 SY_PD_CYCLE_SLIP_LD_FT_EN_TX_HI	Assertion time setting for SY_PD_CYCLE_SLIP_LD_FT_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_CYCLE_SLIP_LD_FT_EN signal or group will transition from LO to HI.

45.3.5.3.1.27 TSM_TIMING15 (TIMING15)

45.3.5.3.1.27.1 Offset

Register	Offset
TIMING15	6Ch

45.3.5.3.1.27.2 Diagram

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

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

45.3.5.3.1.27.3 Fields

Field	Function
31-24 SY_VCO_EN_RX_LO	De-assertion time setting for SY_VCO_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_VCO_EN signal or group will transition from HI to LO.
23-16	Assertion time setting for SY_VCO_EN (RX)

Table continues on the next page...

Field	Function
SY_VCO_EN_RX_HI	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_VCO_EN signal or group will transition from LO to HI.
15-8 SY_VCO_EN_TX_LO	De-assertion time setting for SY_VCO_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_VCO_EN signal or group will transition from HI to LO.
7-0 SY_VCO_EN_TX_HI	Assertion time setting for SY_VCO_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_VCO_EN signal or group will transition from LO to HI.

45.3.5.3.1.28 TSM_TIMING16 (TIMING16)

45.3.5.3.1.28.1 Offset

Register	Offset
TIMING16	70h

45.3.5.3.1.28.2 Diagram

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

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

45.3.5.3.1.28.3 Fields

Field	Function
31-24 SY_LO_RX_BUF_EN_RX_LO	De-assertion time setting for SY_LO_RX_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_RX_BUF_EN signal or group will transition from HI to LO.
23-16 SY_LO_RX_BUF_EN_RX_HI	Assertion time setting for SY_LO_RX_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_RX_BUF_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.29 TSM_TIMING17 (TIMING17)

45.3.5.3.1.29.1 Offset

Register	Offset
TIMING17	74h

45.3.5.3.1.29.2 Diagram

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

45.3.5.3.1.29.3 Fields

Field	Function
31-16 —	Reserved
15-8 SY_LO_TX_BUF_EN_TX_LO	De-assertion time setting for SY_LO_TX_BUF_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_TX_BUF_EN signal or group will transition from HI to LO.
7-0 SY_LO_TX_BUF_EN_TX_HI	Assertion time setting for SY_LO_TX_BUF_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_TX_BUF_EN signal or group will transition from LO to HI.

45.3.5.3.1.30 TSM_TIMING18 (TIMING18)

45.3.5.3.1.30.1 Offset

Register	Offset
TIMING18	78h

45.3.5.3.1.30.2 Diagram

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

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

45.3.5.3.1.30.3 Fields

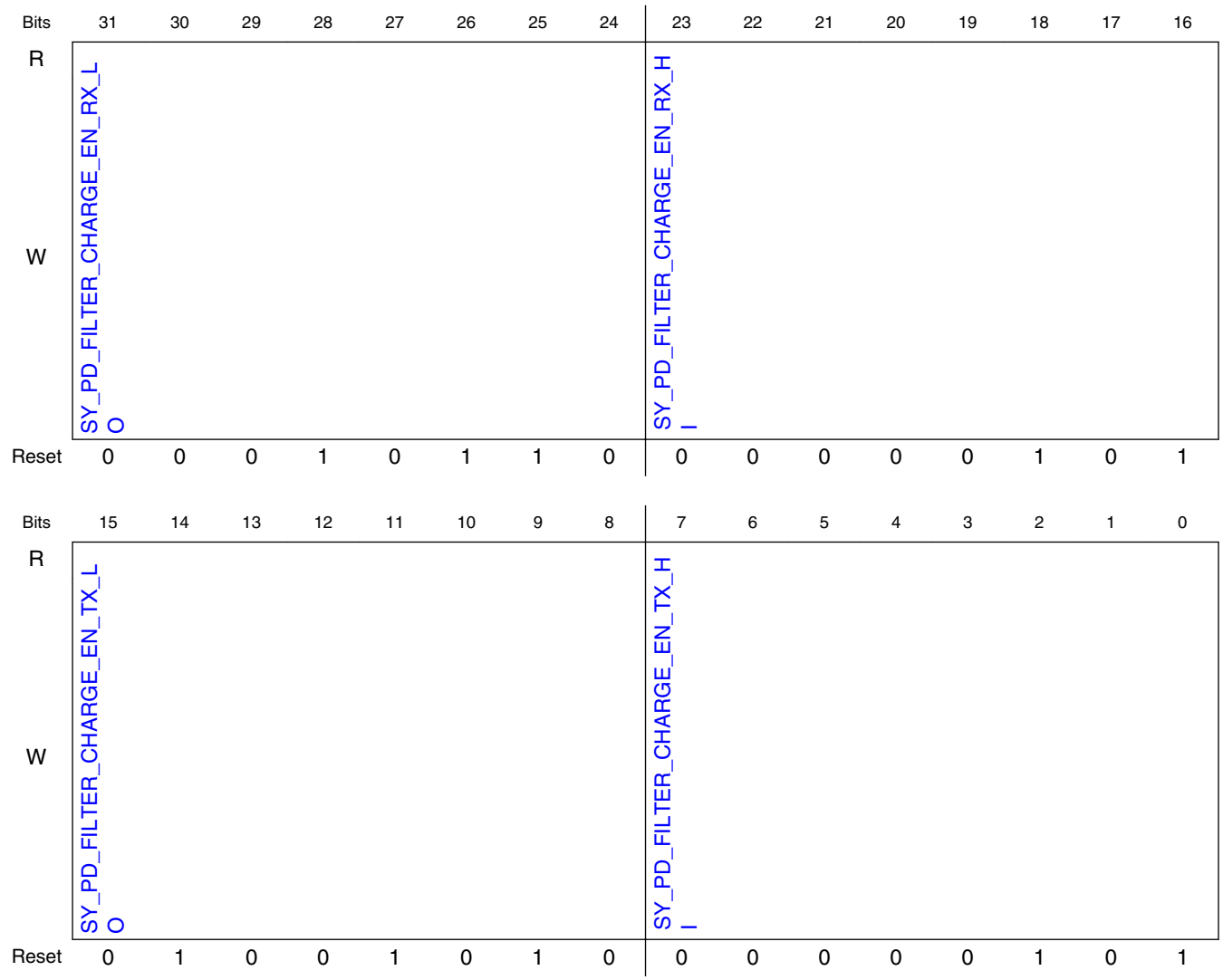
Field	Function
31-24 SY_DIVN_EN_RX_LO	De-assertion time setting for SY_DIVN_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_DIVN_EN signal or group will transition from HI to LO.
23-16 SY_DIVN_EN_RX_HI	Assertion time setting for SY_DIVN_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_DIVN_EN signal or group will transition from LO to HI.
15-8 SY_DIVN_EN_TX_LO	De-assertion time setting for SY_DIVN_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_DIVN_EN signal or group will transition from HI to LO.
7-0 SY_DIVN_EN_TX_HI	Assertion time setting for SY_DIVN_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_DIVN_EN signal or group will transition from LO to HI.

45.3.5.3.1.31 TSM_TIMING19 (TIMING19)

45.3.5.3.1.31.1 Offset

Register	Offset
TIMING19	7Ch

45.3.5.3.1.31.2 Diagram



45.3.5.3.1.31.3 Fields

Field	Function
31-24 SY_PD_FILTER_CHARGE_EN_RX_LO	De-assertion time setting for SY_PD_FILTER_CHARGE_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_FILTER_CHARGE_EN signal or group will transition from HI to LO.
23-16 SY_PD_FILTER_CHARGE_EN_RX_HI	Assertion time setting for SY_PD_FILTER_CHARGE_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_FILTER_CHARGE_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for SY_PD_FILTER_CHARGE_EN (TX)

Table continues on the next page...

Field	Function
SY_PD_FILTER_CHARGE_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_FILTER_CHARGE_EN signal or group will transition from HI to LO.
7-0 SY_PD_FILTER_CHARGE_EN_TX_HI	Assertion time setting for SY_PD_FILTER_CHARGE_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_FILTER_CHARGE_EN signal or group will transition from LO to HI.

45.3.5.3.1.32 TSM_TIMING20 (TIMING20)

45.3.5.3.1.32.1 Offset

Register	Offset
TIMING20	80h

45.3.5.3.1.32.2 Diagram

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

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

45.3.5.3.1.32.3 Fields

Field	Function
31-24 SY_PD_EN_RX_LO	De-assertion time setting for SY_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_EN signal or group will transition from HI to LO.
23-16 SY_PD_EN_RX_HI	Assertion time setting for SY_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_EN signal or group will transition from LO to HI.
15-8 SY_PD_EN_TX_LO	De-assertion time setting for SY_PD_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_EN signal or group will transition from HI to LO.

Table continues on the next page...

Field	Function
7-0 SY_PD_EN_TX_HI	Assertion time setting for SY_PD_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_EN signal or group will transition from LO to HI.

45.3.5.3.1.33 TSM_TIMING21 (TIMING21)

45.3.5.3.1.33.1 Offset

Register	Offset
TIMING21	84h

45.3.5.3.1.33.2 Diagram

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

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

45.3.5.3.1.33.3 Fields

Field	Function
31-24 SY_LO_DIVN_EN_RX_LO	De-assertion time setting for SY_LO_DIVN_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_DIVN_EN signal or group will transition from HI to LO.
23-16 SY_LO_DIVN_EN_RX_HI	Assertion time setting for SY_LO_DIVN_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_DIVN_EN signal or group will transition from LO to HI.
15-8 SY_LO_DIVN_EN_TX_LO	De-assertion time setting for SY_LO_DIVN_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_DIVN_EN signal or group will transition from HI to LO.
7-0 SY_LO_DIVN_EN_TX_HI	Assertion time setting for SY_LO_DIVN_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_DIVN_EN signal or group will transition from LO to HI.

45.3.5.3.1.34 TSM_TIMING22 (TIMING22)

45.3.5.3.1.34.1 Offset

Register	Offset
TIMING22	88h

45.3.5.3.1.34.2 Diagram

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

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

45.3.5.3.1.34.3 Fields

Field	Function
31-24 SY_LO_RX_EN_RX_LO	De-assertion time setting for SY_LO_RX_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_RX_EN signal or group will transition from HI to LO.
23-16 SY_LO_RX_EN_RX_HI	Assertion time setting for SY_LO_RX_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_RX_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.35 TSM_TIMING23 (TIMING23)

45.3.5.3.1.35.1 Offset

Register	Offset
TIMING23	8Ch

45.3.5.3.1.35.2 Diagram

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

45.3.5.3.1.35.3 Fields

Field	Function
31-16 —	Reserved
15-8 SY_LO_TX_EN_TX_LO	De-assertion time setting for SY_LO_TX_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_TX_EN signal or group will transition from HI to LO.
7-0 SY_LO_TX_EN_TX_HI	Assertion time setting for SY_LO_TX_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_TX_EN signal or group will transition from LO to HI.

45.3.5.3.1.36 TSM_TIMING24 (TIMING24)

45.3.5.3.1.36.1 Offset

Register	Offset
TIMING24	90h

45.3.5.3.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SY_DIVN_CAL_EN_RX_LO								SY_DIVN_CAL_EN_RX_HI							
Reset	0	0	0	1	0	1	1	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	SY_DIVN_CAL_EN_TX_LO								SY_DIVN_CAL_EN_TX_HI							
Reset	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

45.3.5.3.1.36.3 Fields

Field	Function
31-24 SY_DIVN_CAL_EN_RX_LO	De-assertion time setting for SY_DIVN_CAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_DIVN_CAL_EN signal or group will transition from HI to LO.
23-16 SY_DIVN_CAL_EN_RX_HI	Assertion time setting for SY_DIVN_CAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_DIVN_CAL_EN signal or group will transition from LO to HI.
15-8 SY_DIVN_CAL_EN_TX_LO	De-assertion time setting for SY_DIVN_CAL_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_DIVN_CAL_EN signal or group will transition from HI to LO.
7-0 SY_DIVN_CAL_EN_TX_HI	Assertion time setting for SY_DIVN_CAL_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_DIVN_CAL_EN signal or group will transition from LO to HI.

45.3.5.3.1.37 TSM_TIMING25 (TIMING25)

45.3.5.3.1.37.1 Offset

Register	Offset
TIMING25	94h

45.3.5.3.1.37.2 Diagram

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

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

45.3.5.3.1.37.3 Fields

Field	Function
31-24 RX_LNA_MIXER_EN_RX_LO	De-assertion time setting for RX_LNA_MIXER_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_LNA_MIXER_EN signal or group will transition from HI to LO.
23-16 RX_LNA_MIXER_EN_RX_HI	Assertion time setting for RX_LNA_MIXER_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_LNA_MIXER_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.38 TSM_TIMING26 (TIMING26)

45.3.5.3.1.38.1 Offset

Register	Offset
TIMING26	98h

45.3.5.3.1.38.2 Diagram

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

45.3.5.3.1.38.3 Fields

Field	Function
31-16 —	Reserved
15-8 TX_PA_EN_TX_LO	De-assertion time setting for TX_PA_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_PA_EN signal or group will transition from HI to LO.
7-0 TX_PA_EN_TX_HI	Assertion time setting for TX_PA_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_PA_EN signal or group will transition from LO to HI.

45.3.5.3.1.39 TSM_TIMING27 (TIMING27)

45.3.5.3.1.39.1 Offset

Register	Offset
TIMING27	9Ch

45.3.5.3.1.39.2 Diagram

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

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

45.3.5.3.1.39.3 Fields

Field	Function
31-24 RX_ADC_I_Q_EN_RX_LO	De-assertion time setting for RX_ADC_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_ADC_I_Q_EN signal or group will transition from HI to LO.
23-16 RX_ADC_I_Q_EN_RX_HI	Assertion time setting for RX_ADC_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_ADC_I_Q_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.40 TSM_TIMING28 (TIMING28)

45.3.5.3.1.40.1 Offset

Register	Offset
TIMING28	A0h

45.3.5.3.1.40.2 Diagram

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

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

45.3.5.3.1.40.3 Fields

Field	Function
31-24 RX_ADC_RESET_EN_RX_LO	De-assertion time setting for RX_ADC_RESET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_ADC_RESET_EN signal or group will transition from HI to LO.
23-16 RX_ADC_RESET_EN_RX_HI	Assertion time setting for RX_ADC_RESET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_ADC_RESET_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.41 TSM_TIMING29 (TIMING29)

45.3.5.3.1.41.1 Offset

Register	Offset
TIMING29	A4h

45.3.5.3.1.41.2 Diagram

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

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

45.3.5.3.1.41.3 Fields

Field	Function
31-24 RX_BBA_I_Q_EN_RX_LO	De-assertion time setting for RX_BBA_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_I_Q_EN signal or group will transition from HI to LO.
23-16 RX_BBA_I_Q_EN_RX_HI	Assertion time setting for RX_BBA_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_I_Q_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.42 TSM_TIMING30 (TIMING30)

45.3.5.3.1.42.1 Offset

Register	Offset
TIMING30	A8h

45.3.5.3.1.42.2 Diagram

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

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

45.3.5.3.1.42.3 Fields

Field	Function
31-24 RX_BBA_PDET_EN_RX_LO	De-assertion time setting for RX_BBA_PDET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_PDET_EN signal or group will transition from HI to LO.
23-16 RX_BBA_PDET_EN_RX_HI	Assertion time setting for RX_BBA_PDET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_PDET_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.43 TSM_TIMING31 (TIMING31)

45.3.5.3.1.43.1 Offset

Register	Offset
TIMING31	ACh

45.3.5.3.1.43.2 Diagram

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

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

45.3.5.3.1.43.3 Fields

Field	Function
31-24 RX_BBA_TZA_DCOC_EN_RX_LO	De-assertion time setting for RX_BBA_TZA_DCOC_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_TZA_DCOC_EN signal or group will transition from HI to LO.
23-16 RX_BBA_TZA_DCOC_EN_RX_HI	Assertion time setting for RX_BBA_TZA_DCOC_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_TZA_DCOC_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.44 TSM_TIMING32 (TIMING32)

45.3.5.3.1.44.1 Offset

Register	Offset
TIMING32	B0h

45.3.5.3.1.44.2 Diagram

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

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

45.3.5.3.1.44.3 Fields

Field	Function
31-24 RX_TZA_I_Q_EN_RX_LO	De-assertion time setting for RX_TZA_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_TZA_I_Q_EN signal or group will transition from HI to LO.
23-16 RX_TZA_I_Q_EN_RX_HI	Assertion time setting for RX_TZA_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_TZA_I_Q_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.45 TSM_TIMING33 (TIMING33)

45.3.5.3.1.45.1 Offset

Register	Offset
TIMING33	B4h

45.3.5.3.1.45.2 Diagram

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

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

45.3.5.3.1.45.3 Fields

Field	Function
31-24 RX_TZA_PDET_EN_RX_LO	De-assertion time setting for RX_TZA_PDET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_TZA_PDET_EN signal or group will transition from HI to LO.
23-16 RX_TZA_PDET_EN_RX_HI	Assertion time setting for RX_TZA_PDET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_TZA_PDET_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.46 TSM_TIMING34 (TIMING34)

45.3.5.3.1.46.1 Offset

Register	Offset
TIMING34	B8h

45.3.5.3.1.46.2 Diagram

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

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

45.3.5.3.1.46.3 Fields

Field	Function
31-24 PLL_DIG_EN_RX_LO	De-assertion time setting for PLL_DIG_EN (RX) 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 (RX) 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	De-assertion time setting for PLL_DIG_EN (TX) 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.
7-0 PLL_DIG_EN_TX_HI	Assertion time setting for PLL_DIG_EN (TX) 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.

45.3.5.3.1.47 TSM_TIMING35 (TIMING35)

45.3.5.3.1.47.1 Offset

Register	Offset
TIMING35	BCh

45.3.5.3.1.47.2 Diagram

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

45.3.5.3.1.47.3 Fields

Field	Function
31-16 —	Reserved
15-8 TX_DIG_EN_TX_LO	De-assertion time setting for TX_DIG_EN (TX) 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.
7-0 TX_DIG_EN_TX_HI	Assertion time setting for TX_DIG_EN (TX) 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.

45.3.5.3.1.48 TSM_TIMING36 (TIMING36)

45.3.5.3.1.48.1 Offset

Register	Offset
TIMING36	C0h

45.3.5.3.1.48.2 Diagram

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

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

45.3.5.3.1.48.3 Fields

Field	Function
31-24 RX_DIG_EN_RX_LO	De-assertion time setting for RX_DIG_EN (RX) 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 (RX) 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.
15-0 —	Reserved

45.3.5.3.1.49 TSM_TIMING37 (TIMING37)

45.3.5.3.1.49.1 Offset

Register	Offset
TIMING37	C4h

45.3.5.3.1.49.2 Diagram

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

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

45.3.5.3.1.49.3 Fields

Field	Function
31-24 RX_INIT_RX_LO	De-assertion time setting for RX_INIT (RX) 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 (RX) 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.
15-0 —	Reserved

45.3.5.3.1.50 TSM_TIMING38 (TIMING38)

45.3.5.3.1.50.1 Offset

Register	Offset
TIMING38	C8h

45.3.5.3.1.50.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SIGMA_DELTA_EN_RX_LO								SIGMA_DELTA_EN_RX_HI							
Reset	0	1	1	0	0	1	1	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	SIGMA_DELTA_EN_TX_LO								SIGMA_DELTA_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0

45.3.5.3.1.50.3 Fields

Field	Function
31-24 SIGMA_DELTA_EN_RX_LO	De-assertion time setting for SIGMA_DELTA_EN (RX) 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 (RX) 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	De-assertion time setting for SIGMA_DELTA_EN (TX) 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.
7-0 SIGMA_DELTA_EN_TX_HI	Assertion time setting for SIGMA_DELTA_EN (TX) 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.

45.3.5.3.1.51 TSM_TIMING39 (TIMING39)

45.3.5.3.1.51.1 Offset

Register	Offset
TIMING39	CCh

45.3.5.3.1.51.2 Diagram

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

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

45.3.5.3.1.51.3 Fields

Field	Function
31-24 RX_PHY_EN_RX_LO	De-assertion time setting for RX_PHY_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_PHY_EN signal or group will transition from HI to LO.
23-16 RX_PHY_EN_RX_HI	Assertion time setting for RX_PHY_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_PHY_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.52 TSM_TIMING40 (TIMING40)

45.3.5.3.1.52.1 Offset

Register	Offset
TIMING40	D0h

45.3.5.3.1.52.2 Diagram

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

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

45.3.5.3.1.52.3 Fields

Field	Function
31-24 DCOC_EN_RX_LO	De-assertion time setting for DCOC_EN (RX) 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 (RX) 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.
15-0 —	Reserved

45.3.5.3.1.53 TSM_TIMING41 (TIMING41)

45.3.5.3.1.53.1 Offset

Register	Offset
TIMING41	D4h

45.3.5.3.1.53.2 Diagram

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

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

45.3.5.3.1.53.3 Fields

Field	Function
31-24 DCOC_INIT_RX_LO	De-assertion time setting for DCOC_INIT (RX) 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 (RX) 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.
15-0 —	Reserved

45.3.5.3.1.54 TSM_TIMING42 (TIMING42)

45.3.5.3.1.54.1 Offset

Register	Offset
TIMING42	D8h

45.3.5.3.1.54.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SAR_ADC_TRIG_EN_RX_LO								SAR_ADC_TRIG_EN_RX_HI							
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	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

45.3.5.3.1.54.3 Fields

Field	Function
31-24 SAR_ADC_TRIG_EN_RX_LO	De-assertion time setting for SAR_ADC_TRIG_EN (RX) 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 (RX) 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	De-assertion time setting for SAR_ADC_TRIG_EN (TX) 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.
7-0 SAR_ADC_TRIG_EN_TX_HI	Assertion time setting for SAR_ADC_TRIG_EN (TX) 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.

45.3.5.3.1.55 TSM_TIMING43 (TIMING43)

45.3.5.3.1.55.1 Offset

Register	Offset
TIMING43	DCh

45.3.5.3.1.55.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSM_SPARE0_EN_RX_LO								TSM_SPARE0_EN_RX_HI							
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	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

45.3.5.3.1.55.3 Fields

Field	Function
31-24 TSM_SPARE0_EN_RX_LO	De-assertion time setting for TSM_SPARE0_EN (RX) 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 (RX) 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	De-assertion time setting for TSM_SPARE0_EN (TX) 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.
7-0 TSM_SPARE0_EN_TX_HI	Assertion time setting for TSM_SPARE0_EN (TX) 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.

45.3.5.3.1.56 TSM_TIMING44 (TIMING44)

45.3.5.3.1.56.1 Offset

Register	Offset
TIMING44	E0h

45.3.5.3.1.56.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSM_SPARE1_EN_RX_LO								TSM_SPARE1_EN_RX_HI							
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	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

45.3.5.3.1.56.3 Fields

Field	Function
31-24 TSM_SPARE1_EN_RX_LO	De-assertion time setting for TSM_SPARE1_EN (RX) 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 (RX) 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	De-assertion time setting for TSM_SPARE1_EN (TX) 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.
7-0 TSM_SPARE1_EN_TX_HI	Assertion time setting for TSM_SPARE1_EN (TX) 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.

45.3.5.3.1.57 TSM_TIMING45 (TIMING45)

45.3.5.3.1.57.1 Offset

Register	Offset
TIMING45	E4h

45.3.5.3.1.57.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSM_SPARE2_EN_RX_LO								TSM_SPARE2_EN_RX_HI							
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	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

45.3.5.3.1.57.3 Fields

Field	Function
31-24 TSM_SPARE2_EN_RX_LO	De-assertion time setting for TSM_SPARE2_EN (RX) 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 (RX) 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	De-assertion time setting for TSM_SPARE2_EN (TX) 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.
7-0 TSM_SPARE2_EN_TX_HI	Assertion time setting for TSM_SPARE2_EN (TX) 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.

45.3.5.3.1.58 TSM_TIMING46 (TIMING46)

45.3.5.3.1.58.1 Offset

Register	Offset
TIMING46	E8h

45.3.5.3.1.58.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSM_SPARE3_EN_RX_LO								TSM_SPARE3_EN_RX_HI							
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	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

45.3.5.3.1.58.3 Fields

Field	Function
31-24 TSM_SPARE3_EN_RX_LO	De-assertion time setting for TSM_SPARE3_EN (RX) 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 (RX) 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	De-assertion time setting for TSM_SPARE3_EN (TX) 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.
7-0 TSM_SPARE3_EN_TX_HI	Assertion time setting for TSM_SPARE3_EN (TX) 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.

45.3.5.3.1.59 TSM_TIMING47 (TIMING47)

45.3.5.3.1.59.1 Offset

Register	Offset
TIMING47	ECh

45.3.5.3.1.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO0_TRIG_EN_RX_LO								GPIO0_TRIG_EN_RX_HI							
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	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

45.3.5.3.1.59.3 Fields

Field	Function
31-24 GPIO0_TRIG_EN_RX_LO	De-assertion time setting for GPIO0_TRIG_EN (RX) 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 (RX) 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	De-assertion time setting for GPIO0_TRIG_EN (TX) 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.
7-0 GPIO0_TRIG_EN_TX_HI	Assertion time setting for GPIO0_TRIG_EN (TX) 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.

45.3.5.3.1.60 TSM_TIMING48 (TIMING48)

45.3.5.3.1.60.1 Offset

Register	Offset
TIMING48	F0h

45.3.5.3.1.60.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO1_TRIG_EN_RX_LO								GPIO1_TRIG_EN_RX_HI							
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	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

45.3.5.3.1.60.3 Fields

Field	Function
31-24 GPIO1_TRIG_EN_RX_LO	De-assertion time setting for GPIO1_TRIG_EN (RX) 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 (RX) 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	De-assertion time setting for GPIO1_TRIG_EN (TX) 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.
7-0 GPIO1_TRIG_EN_TX_HI	Assertion time setting for GPIO1_TRIG_EN (TX) 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.

45.3.5.3.1.61 TSM_TIMING49 (TIMING49)

45.3.5.3.1.61.1 Offset

Register	Offset
TIMING49	F4h

45.3.5.3.1.61.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO2_TRIG_EN_RX_LO								GPIO2_TRIG_EN_RX_HI							
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	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

45.3.5.3.1.61.3 Fields

Field	Function
31-24 GPIO2_TRIG_EN_RX_LO	De-assertion time setting for GPIO2_TRIG_EN (RX) 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 (RX) 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	De-assertion time setting for GPIO2_TRIG_EN (TX) 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.
7-0 GPIO2_TRIG_EN_TX_HI	Assertion time setting for GPIO2_TRIG_EN (TX) 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.

45.3.5.3.1.62 TSM_TIMING50 (TIMING50)

45.3.5.3.1.62.1 Offset

Register	Offset
TIMING50	F8h

45.3.5.3.1.62.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO3_TRIG_EN_RX_LO								GPIO3_TRIG_EN_RX_HI							
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	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

45.3.5.3.1.62.3 Fields

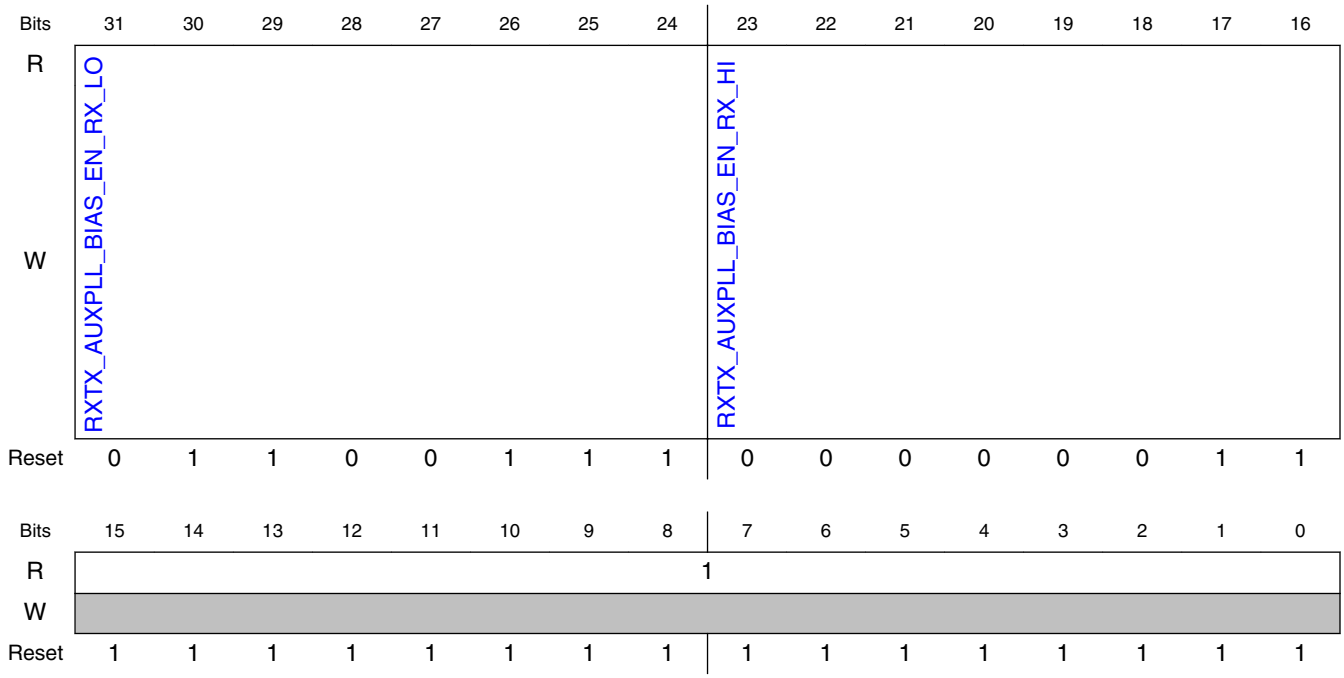
Field	Function
31-24 GPIO3_TRIG_EN_RX_LO	De-assertion time setting for GPIO3_TRIG_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the GPIO3_TRIG_EN signal or group will transition from HI to LO.
23-16 GPIO3_TRIG_EN_RX_HI	Assertion time setting for GPIO3_TRIG_EN (RX) 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	De-assertion time setting for GPIO3_TRIG_EN (TX) 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.
7-0 GPIO3_TRIG_EN_TX_HI	Assertion time setting for GPIO3_TRIG_EN (TX) 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.

45.3.5.3.1.63 TSM_TIMING51 (TIMING51)

45.3.5.3.1.63.1 Offset

Register	Offset
TIMING51	FCh

45.3.5.3.1.63.2 Diagram



45.3.5.3.1.63.3 Fields

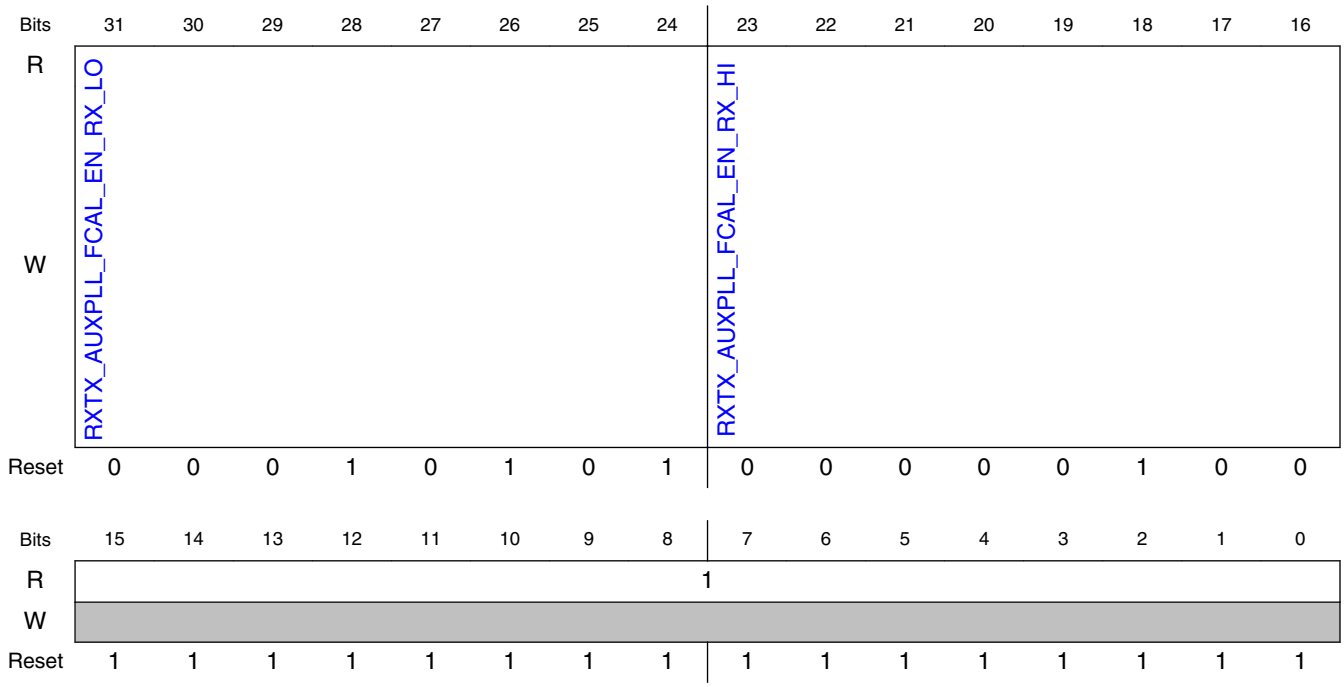
Field	Function
31-24 RXTX_AUXPLL_BIAS_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_BIAS_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_BIAS_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_BIAS_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_BIAS_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_BIAS_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.64 TSM_TIMING52 (TIMING52)

45.3.5.3.1.64.1 Offset

Register	Offset
TIMING52	100h

45.3.5.3.1.64.2 Diagram



45.3.5.3.1.64.3 Fields

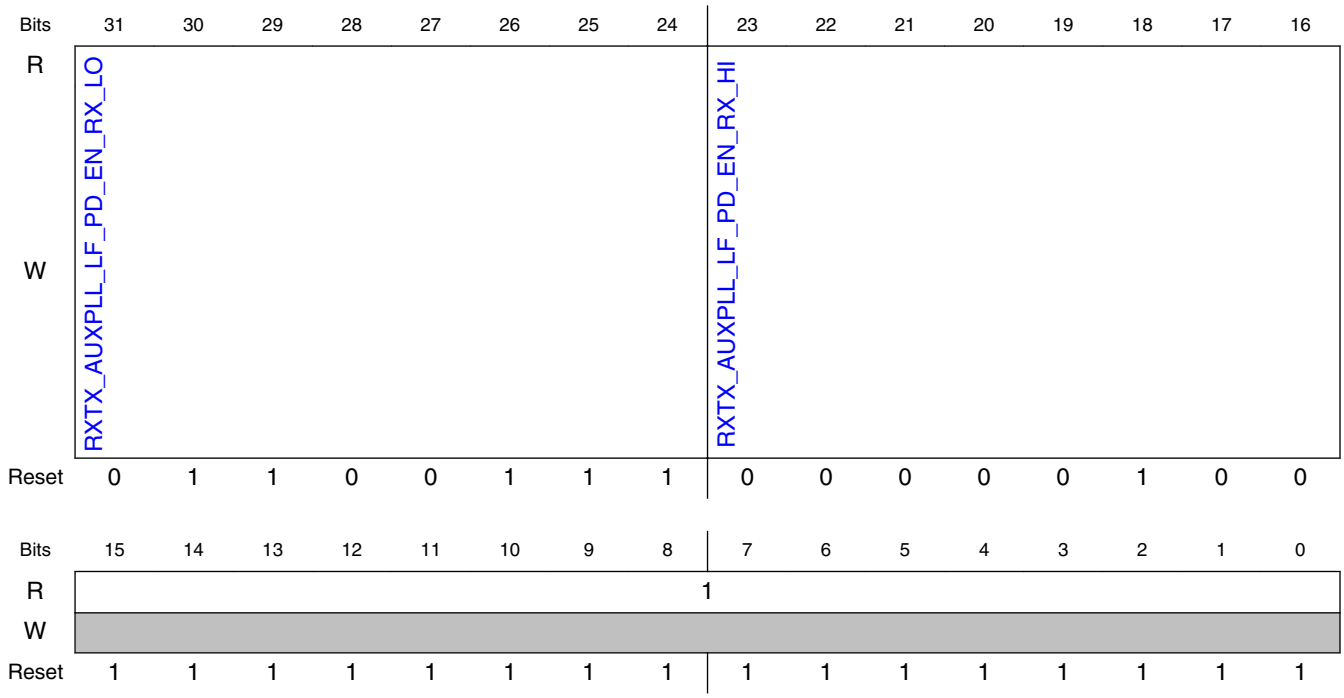
Field	Function
31-24 RXTX_AUXPLL_FCAL_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_FCAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_FCAL_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_FCAL_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_FCAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_FCAL_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.65 TSM_TIMING53 (TIMING53)

45.3.5.3.1.65.1 Offset

Register	Offset
TIMING53	104h

45.3.5.3.1.65.2 Diagram



45.3.5.3.1.65.3 Fields

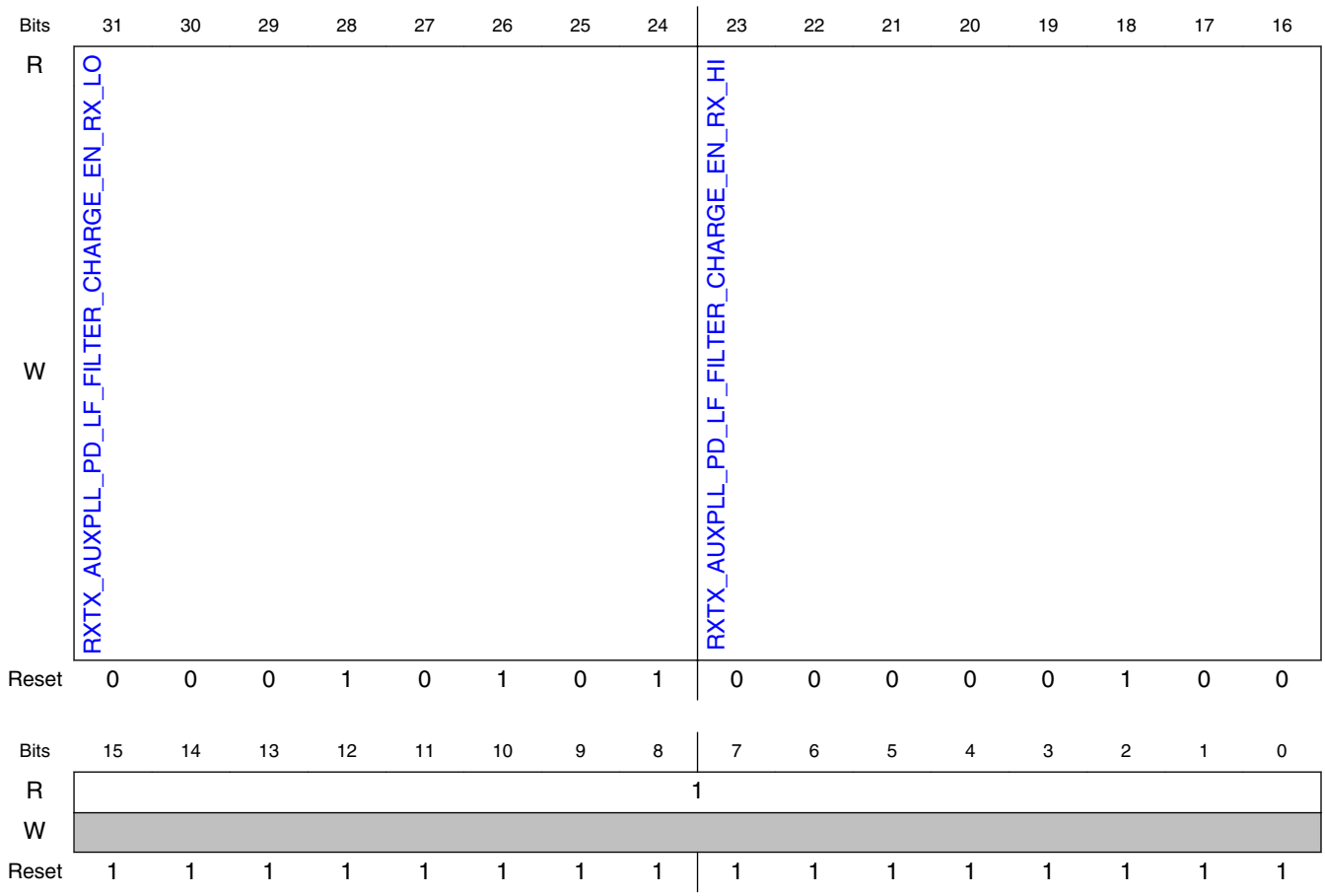
Field	Function
31-24 RXTX_AUXPLL_LF_PD_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_LF_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_LF_PD_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_LF_PD_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_LF_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_LF_PD_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.66 TSM_TIMING54 (TIMING54)

45.3.5.3.1.66.1 Offset

Register	Offset
TIMING54	108h

45.3.5.3.1.66.2 Diagram



45.3.5.3.1.66.3 Fields

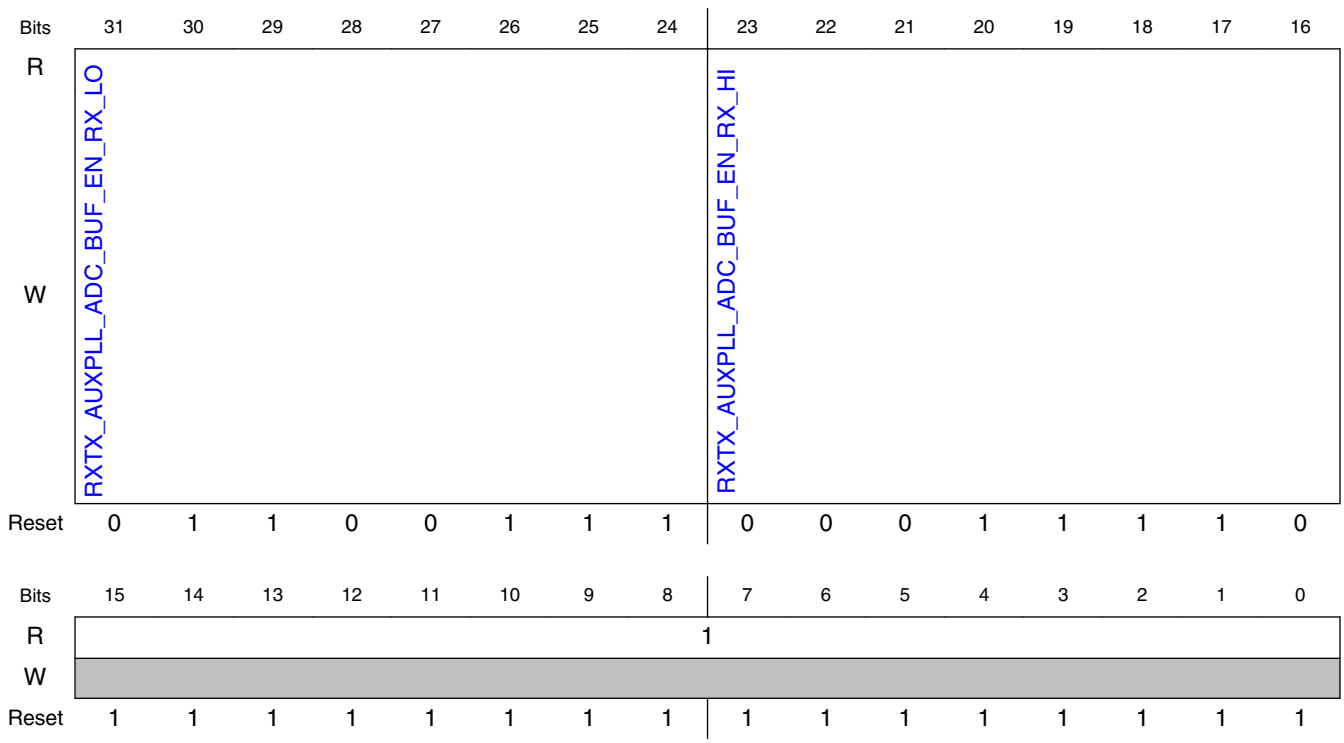
Field	Function
31-24 RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.67 TSM_TIMING55 (TIMING55)

45.3.5.3.1.67.1 Offset

Register	Offset
TIMING55	10Ch

45.3.5.3.1.67.2 Diagram



45.3.5.3.1.67.3 Fields

Field	Function
31-24 RXTX_AUXPLL_ADC_BUF_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_ADC_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_ADC_BUF_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_ADC_BUF_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_ADC_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_ADC_BUF_EN signal or group will transition from LO to HI.

Table continues on the next page...

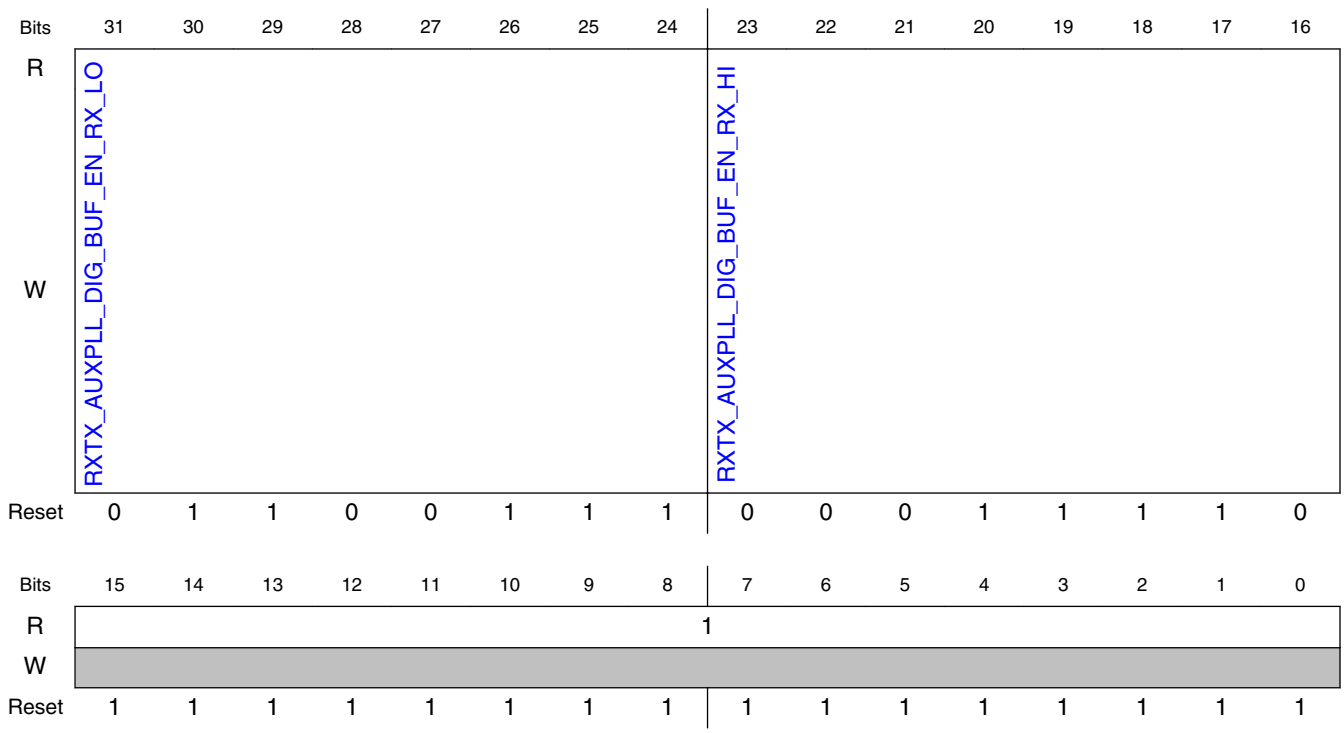
Field	Function
15-0 —	Reserved

45.3.5.3.1.68 TSM_TIMING56 (TIMING56)

45.3.5.3.1.68.1 Offset

Register	Offset
TIMING56	110h

45.3.5.3.1.68.2 Diagram



45.3.5.3.1.68.3 Fields

Field	Function
31-24	De-assertion time setting for RXTX_AUXPLL_DIG_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_DIG_BUF_EN signal or group will transition from HI to LO.

Table continues on the next page...

Field	Function
RXTX_AUXPLL_DIG_BUF_EN_RX_LO	
23-16 RXTX_AUXPLL_DIG_BUF_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_DIG_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_DIG_BUF_EN signal or group will transition from LO to HI.
15-0 —	Reserved

45.3.5.3.1.69 TSM_TIMING57 (TIMING57)

45.3.5.3.1.69.1 Offset

Register	Offset
TIMING57	114h

45.3.5.3.1.69.2 Diagram

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

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

45.3.5.3.1.69.3 Fields

Field	Function
31-24 RXTX_RCCAL_EN_RX_LO	De-assertion time setting for RXTX_RCCAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_RCCAL_EN signal or group will transition from HI to LO.
23-16 RXTX_RCCAL_EN_RX_HI	Assertion time setting for RXTX_RCCAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_RCCAL_EN signal or group will transition from LO to HI.

Table continues on the next page...

Field	Function
15-0 —	Reserved

45.3.5.3.1.70 TSM_TIMING58 (TIMING58)

45.3.5.3.1.70.1 Offset

Register	Offset
TIMING58	118h

45.3.5.3.1.70.2 Diagram

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

45.3.5.3.1.70.3 Fields

Field	Function
31-16 —	Reserved
15-8 TX_HPM_DAC_EN_TX_LO	De-assertion time setting for TX_HPM_DAC_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_HPM_DAC_EN signal or group will transition from HI to LO.
7-0 TX_HPM_DAC_EN_TX_HI	Assertion time setting for TX_HPM_DAC_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_HPM_DAC_EN signal or group will transition from LO to HI.

45.3.5.3.1.71 TSM OVERRIDE REGISTER 0 (OVRD0)

45.3.5.3.1.71.1 Offset

Register	Offset
OVRD0	11Ch

45.3.5.3.1.71.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	SY_VCO_EN_OVRD	SY_VCO_EN_OVRD_EN	SY_PD_CYCLE_SLIP_LD_EN_OVRD	SY_PD_CYCLE_SLIP_LD_EN_OVRD_EN	PLL_LOOP_IS_OPEN_OVRD	PLL_LOOP_IS_OPEN_OVRD_EN	BB_XTAL_AUXPLL_REF_CLK_EN_OVRD	BB_XTAL_AUXPLL_REF_CLK_EN_OVRD_EN	BB_XTAL_DAC_REF_CLK_EN_OVRD	BB_XTAL_DAC_REF_CLK_EN_OVRD_EN	BB_XTAL_PLL_REF_CLK_EN_OVRD	BB_XTAL_PLL_REF_CLK_EN_OVRD_EN	BB_LDO_VCOLO_FASTCHARGE_EN_OVRD	BB_LDO_VCOLO_FASTCHARGE_EN_OVRD_EN	BB_LDO_VCOLO_BLEED_EN_OVRD	BB_LDO_VCOLO_BLEED_EN_OVRD_EN
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	BB_LDO_FDBK_BLEED_EN_OVRD	BB_LDO_FDBK_BLEED_EN_OVRD_EN	BB_LDO_VTREF_EN_OVRD	BB_LDO_VTREF_EN_OVRD_EN	BB_LDO_VCOLO_EN_OVRD	BB_LDO_VCOLO_EN_OVRD_EN	BB_LDO_FDBK_EN_OVRD	BB_LDO_FDBK_EN_OVRD_EN	BB_LDO_PD_EN_OVRD	BB_LDO_PD_EN_OVRD_EN	BB_LDO_BBA_EN_OVRD	BB_LDO_BBA_EN_OVRD_EN	BB_LDO_ADCDAC_EN_OVRD	BB_LDO_ADCDAC_EN_OVRD_EN	BB_LDO_HF_EN_OVRD	BB_LDO_HF_EN_OVRD_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.5.3.1.71.3 Fields

Field	Function
31 SY_VCO_EN_OVRD	Override value for SY_VCO_EN When SY_VCO_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_vco_en". This bit is ignored when SY_VCO_EN_OVRD_EN==0.
30 SY_VCO_EN_OVRD_EN	Override control for SY_VCO_EN 0b - Normal operation. 1b - Use the state of SY_VCO_EN_OVRD to override the signal "sy_vco_en".
29 SY_PD_CYCLE_SLIP_LD_EN_OVRD	Override value for SY_PD_CYCLE_SLIP_LD_EN When SY_PD_CYCLE_SLIP_LD_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_pd_cycle_slip_ld_en". This bit is ignored when SY_PD_CYCLE_SLIP_LD_EN_OVRD_EN==0.
28 SY_PD_CYCLE_SLIP_LD_EN_OVRD_EN	Override control for SY_PD_CYCLE_SLIP_LD_EN 0b - Normal operation. 1b - Use the state of SY_PD_CYCLE_SLIP_LD_EN_OVRD to override the signal "sy_pd_cycle_slip_ld_en".
27 PLL_LOOP_IS_OPEN_OVRD	Override value for PLL_LOOP_IS_OPEN When PLL_LOOP_IS_OPEN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_loop_is_open". This bit is ignored when PLL_LOOP_IS_OPEN_OVRD_EN==0.
26 PLL_LOOP_IS_OPEN_OVRD_EN	Override control for PLL_LOOP_IS_OPEN 0b - Normal operation. 1b - Use the state of PLL_LOOP_IS_OPEN_OVRD to override the signal "pll_loop_is_open".
25 BB_XTAL_AUXPLL_REF_CLK_EN_OVRD	Override value for BB_XTAL_AUXPLL_REF_CLK_EN When BB_XTAL_AUXPLL_REF_CLK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_xtal_auxpll_ref_clk_en". This bit is ignored when BB_XTAL_AUXPLL_REF_CLK_EN_OVRD_EN==0.
24 BB_XTAL_AUXPLL_REF_CLK_EN_OVRD_EN	Override control for BB_XTAL_AUXPLL_REF_CLK_EN 0b - Normal operation. 1b - Use the state of BB_XTAL_AUXPLL_REF_CLK_EN_OVRD to override the signal "bb_xtal_auxpll_ref_clk_en".
23 BB_XTAL_DAC_REF_CLK_EN_OVRD	Override value for BB_XTAL_DAC_REF_CLK_EN When BB_XTAL_DAC_REF_CLK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_xtal_dac_ref_clk_en". This bit is ignored when BB_XTAL_DAC_REF_CLK_EN_OVRD_EN==0.
22 BB_XTAL_DAC_REF_CLK_EN_OVRD_EN	Override control for BB_XTAL_DAC_REF_CLK_EN 0b - Normal operation. 1b - Use the state of BB_XTAL_DAC_REF_CLK_EN_OVRD to override the signal "bb_xtal_dac_ref_clk_en".
21 BB_XTAL_PLL_REF_CLK_EN_OVRD	Override value for BB_XTAL_PLL_REF_CLK_EN When BB_XTAL_PLL_REF_CLK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_xtal_pll_ref_clk_en". This bit is ignored when BB_XTAL_PLL_REF_CLK_EN_OVRD_EN==0.
20	Override control for BB_XTAL_PLL_REF_CLK_EN 0b - Normal operation.

Table continues on the next page...

Field	Function
BB_XTAL_PLL_REF_CLK_EN_OVRD_EN	1b - Use the state of BB_XTAL_PLL_REF_CLK_EN_OVRD to override the signal "bb_xtal_pll_ref_clk_en".
19 BB_LDO_VCOLO_FASTCHARGE_EN_OVRD	Override value for BB_LDO_VCOLO_FASTCHARGE_EN When BB_LDO_VCOLO_FASTCHARGE_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_vcolo_fastcharge_en". This bit is ignored when BB_LDO_VCOLO_FASTCHARGE_EN_OVRD_EN==0.
18 BB_LDO_VCOLO_FASTCHARGE_EN_OVRD_EN	Override control for BB_LDO_VCOLO_FASTCHARGE_EN 0b - Normal operation. 1b - Use the state of BB_LDO_VCOLO_FASTCHARGE_EN_OVRD to override the signal "bb_ldo_vcolo_fastcharge_en".
17 BB_LDO_VCOLO_BLEED_EN_OVRD	Override value for BB_LDO_VCOLO_BLEED_EN When BB_LDO_VCOLO_BLEED_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_vcolo_bleed_en". This bit is ignored when BB_LDO_VCOLO_BLEED_EN_OVRD_EN==0.
16 BB_LDO_VCOLO_BLEED_EN_OVRD_EN	Override control for BB_LDO_VCOLO_BLEED_EN 0b - Normal operation. 1b - Use the state of BB_LDO_VCOLO_BLEED_EN_OVRD to override the signal "bb_ldo_vcolo_bleed_en".
15 BB_LDO_FDBK_BLEED_EN_OVRD	Override value for BB_LDO_FDBK_BLEED_EN When BB_LDO_FDBK_BLEED_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_fdbk_bleed_en". This bit is ignored when BB_LDO_FDBK_BLEED_EN_OVRD_EN==0.
14 BB_LDO_FDBK_BLEED_EN_OVRD_EN	Override control for BB_LDO_FDBK_BLEED_EN 0b - Normal operation. 1b - Use the state of BB_LDO_FDBK_BLEED_EN_OVRD to override the signal "bb_ldo_fdbk_bleed_en".
13 BB_LDO_VTREF_EN_OVRD	Override value for BB_LDO_VTREF_EN When BB_LDO_VTREF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_vtref_en". This bit is ignored when BB_LDO_VTREF_EN_OVRD_EN==0.
12 BB_LDO_VTREF_EN_OVRD_EN	Override control for BB_LDO_VTREF_EN 0b - Normal operation. 1b - Use the state of BB_LDO_VTREF_EN_OVRD to override the signal "bb_ldo_vtref_en".
11 BB_LDO_VCOLO_EN_OVRD	Override value for BB_LDO_VCOLO_EN When BB_LDO_VCOLO_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_vcolo_en". This bit is ignored when BB_LDO_VCOLO_EN_OVRD_EN==0.
10 BB_LDO_VCOLO_EN_OVRD_EN	Override control for BB_LDO_VCOLO_EN 0b - Normal operation. 1b - Use the state of BB_LDO_VCOLO_EN_OVRD to override the signal "bb_ldo_vcolo_en".
9 BB_LDO_FDBK_EN_OVRD	Override value for BB_LDO_FDBK_EN When BB_LDO_FDBK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_fdbk_en". This bit is ignored when BB_LDO_FDBK_EN_OVRD_EN==0.
8	Override control for BB_LDO_FDBK_EN 0b - Normal operation.

Table continues on the next page...

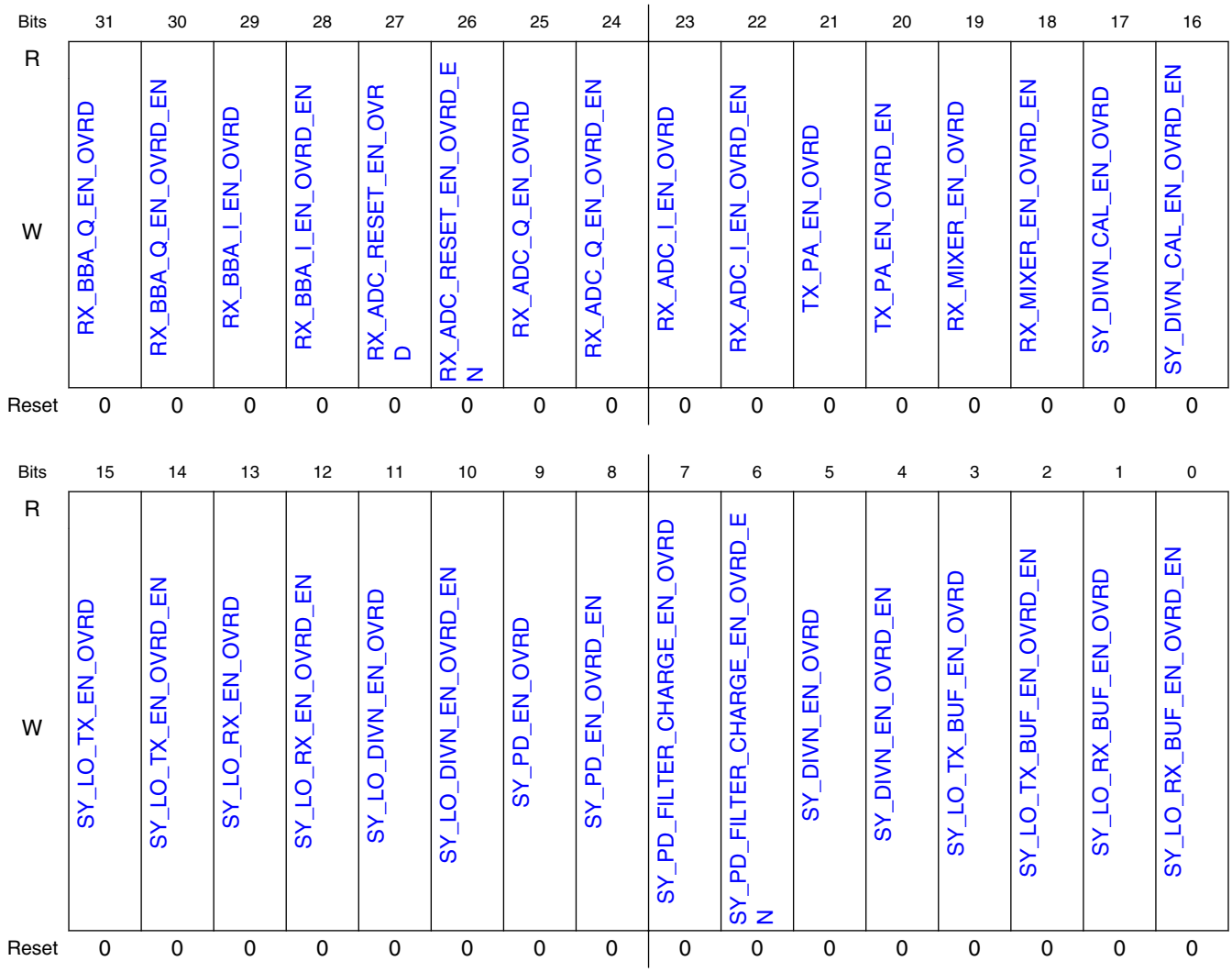
Field	Function
BB_LDO_FDBK_EN_OVRD_EN	1b - Use the state of BB_LDO_FDBK_EN_OVRD to override the signal "bb_ldo_fdbk_en".
7 BB_LDO_PD_EN_OVRD	Override value for BB_LDO_PD_EN When BB_LDO_PD_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_pd_en". This bit is ignored when BB_LDO_PD_EN_OVRD_EN==0.
6 BB_LDO_PD_EN_OVRD_EN	Override control for BB_LDO_PD_EN 0b - Normal operation. 1b - Use the state of BB_LDO_PD_EN_OVRD to override the signal "bb_ldo_pd_en".
5 BB_LDO_BBA_EN_OVRD	Override value for BB_LDO_BBA_EN When BB_LDO_BBA_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_bba_en". This bit is ignored when BB_LDO_BBA_EN_OVRD_EN==0.
4 BB_LDO_BBA_EN_OVRD_EN	Override control for BB_LDO_BBA_EN 0b - Normal operation. 1b - Use the state of BB_LDO_BBA_EN_OVRD to override the signal "bb_ldo_bba_en".
3 BB_LDO_ADCDAC_EN_OVRD	Override value for BB_LDO_ADCDAC_EN When BB_LDO_ADCDAC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_adcdac_en". This bit is ignored when BB_LDO_ADCDAC_EN_OVRD_EN==0.
2 BB_LDO_ADCDAC_EN_OVRD_EN	Override control for BB_LDO_ADCDAC_EN 0b - Normal operation. 1b - Use the state of BB_LDO_ADCDAC_EN_OVRD to override the signal "bb_ldo_adcdac_en".
1 BB_LDO_HF_EN_OVRD	Override value for BB_LDO_HF_EN When BB_LDO_HF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_hf_en". This bit is ignored when BB_LDO_HF_EN_OVRD_EN==0.
0 BB_LDO_HF_EN_OVRD_EN	Override control for BB_LDO_HF_EN 0b - Normal operation. 1b - Use the state of BB_LDO_HF_EN_OVRD to override the signal "bb_ldo_hf_en".

45.3.5.3.1.72 TSM OVERRIDE REGISTER 1 (OVRD1)

45.3.5.3.1.72.1 Offset

Register	Offset
OVRD1	120h

45.3.5.3.1.72.2 Diagram



45.3.5.3.1.72.3 Fields

Field	Function
31 RX_BBA_Q_EN_OVRD	Override value for RX_BBA_Q_EN When RX_BBA_Q_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_bba_q_en". This bit is ignored when RX_BBA_Q_EN_OVRD_EN==0.
30 RX_BBA_Q_EN_OVRD_EN	Override control for RX_BBA_Q_EN 0b - Normal operation. 1b - Use the state of RX_BBA_Q_EN_OVRD to override the signal "rx_bba_q_en".
29 RX_BBA_I_EN_OVRD	Override value for RX_BBA_I_EN When RX_BBA_I_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_bba_i_en". This bit is ignored when RX_BBA_I_EN_OVRD_EN==0.
28	Override control for RX_BBA_I_EN 0b - Normal operation.

Table continues on the next page...

Field	Function
RX_BBA_I_EN_OVRD_EN	1b - Use the state of RX_BBA_I_EN_OVRD to override the signal "rx_bba_i_en".
27 RX_ADC_RESET_EN_OVRD	Override value for RX_ADC_RESET_EN When RX_ADC_RESET_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_adc_reset_en". This bit is ignored when RX_ADC_RESET_EN_OVRD_EN==0.
26 RX_ADC_RESET_EN_OVRD_EN	Override control for RX_ADC_RESET_EN 0b - Normal operation. 1b - Use the state of RX_ADC_RESET_EN_OVRD to override the signal "rx_adc_reset_en".
25 RX_ADC_Q_EN_OVRD	Override value for RX_ADC_Q_EN When RX_ADC_Q_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_adc_q_en". This bit is ignored when RX_ADC_Q_EN_OVRD_EN==0.
24 RX_ADC_Q_EN_OVRD_EN	Override control for RX_ADC_Q_EN 0b - Normal operation. 1b - Use the state of RX_ADC_Q_EN_OVRD to override the signal "rx_adc_q_en".
23 RX_ADC_I_EN_OVRD	Override value for RX_ADC_I_EN When RX_ADC_I_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_adc_i_en". This bit is ignored when RX_ADC_I_EN_OVRD_EN==0.
22 RX_ADC_I_EN_OVRD_EN	Override control for RX_ADC_I_EN 0b - Normal operation. 1b - Use the state of RX_ADC_I_EN_OVRD to override the signal "rx_adc_i_en".
21 TX_PA_EN_OVRD	Override value for TX_PA_EN When TX_PA_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tx_pa_en". This bit is ignored when TX_PA_EN_OVRD_EN==0.
20 TX_PA_EN_OVRD_EN	Override control for TX_PA_EN 0b - Normal operation. 1b - Use the state of TX_PA_EN_OVRD to override the signal "tx_pa_en".
19 RX_MIXER_EN_OVRD	Override value for RX_MIXER_EN When RX_MIXER_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_mixer_en". This bit is ignored when RX_MIXER_EN_OVRD_EN==0.
18 RX_MIXER_EN_OVRD_EN	Override control for RX_MIXER_EN 0b - Normal operation. 1b - Use the state of RX_MIXER_EN_OVRD to override the signal "rx_mixer_en".
17 SY_DIVN_CAL_EN_OVRD	Override value for SY_DIVN_CAL_EN When SY_DIVN_CAL_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_divn_cal_en". This bit is ignored when SY_DIVN_CAL_EN_OVRD_EN==0.
16 SY_DIVN_CAL_EN_OVRD_EN	Override control for SY_DIVN_CAL_EN 0b - Normal operation. 1b - Use the state of SY_DIVN_CAL_EN_OVRD to override the signal "sy_divn_cal_en".
15 SY_LO_TX_EN_OVRD	Override value for SY_LO_TX_EN When SY_LO_TX_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_lo_tx_en". This bit is ignored when SY_LO_TX_EN_OVRD_EN==0.
14 SY_LO_TX_EN_OVRD_EN	Override control for SY_LO_TX_EN 0b - Normal operation. 1b - Use the state of SY_LO_TX_EN_OVRD to override the signal "sy_lo_tx_en".

Table continues on the next page...

Field	Function
13 SY_LO_RX_EN_OVRD	Override value for SY_LO_RX_EN When SY_LO_RX_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_lo_rx_en". This bit is ignored when SY_LO_RX_EN_OVRD_EN==0.
12 SY_LO_RX_EN_OVRD_EN	Override control for SY_LO_RX_EN 0b - Normal operation. 1b - Use the state of SY_LO_RX_EN_OVRD to override the signal "sy_lo_rx_en".
11 SY_LO_DIVN_EN_OVRD	Override value for SY_LO_DIVN_EN When SY_LO_DIVN_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_lo_divn_en". This bit is ignored when SY_LO_DIVN_EN_OVRD_EN==0.
10 SY_LO_DIVN_EN_OVRD_EN	Override control for SY_LO_DIVN_EN 0b - Normal operation. 1b - Use the state of SY_LO_DIVN_EN_OVRD to override the signal "sy_lo_divn_en".
9 SY_PD_EN_OVRD	Override value for SY_PD_EN When SY_PD_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_pd_en". This bit is ignored when SY_PD_EN_OVRD_EN==0.
8 SY_PD_EN_OVRD_EN	Override control for SY_PD_EN 0b - Normal operation. 1b - Use the state of SY_PD_EN_OVRD to override the signal "sy_pd_en".
7 SY_PD_FILTER_CHARGE_EN_OVRD	Override value for SY_PD_FILTER_CHARGE_EN When SY_PD_FILTER_CHARGE_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_pd_filter_charge_en". This bit is ignored when SY_PD_FILTER_CHARGE_EN_OVRD_EN==0.
6 SY_PD_FILTER_CHARGE_EN_OVRD_EN	Override control for SY_PD_FILTER_CHARGE_EN 0b - Normal operation. 1b - Use the state of SY_PD_FILTER_CHARGE_EN_OVRD to override the signal "sy_pd_filter_charge_en".
5 SY_DIVN_EN_OVRD	Override value for SY_DIVN_EN When SY_DIVN_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_divn_en". This bit is ignored when SY_DIVN_EN_OVRD_EN==0.
4 SY_DIVN_EN_OVRD_EN	Override control for SY_DIVN_EN 0b - Normal operation. 1b - Use the state of SY_DIVN_EN_OVRD to override the signal "sy_divn_en".
3 SY_LO_TX_BUF_EN_OVRD	Override value for SY_LO_TX_BUF_EN When SY_LO_TX_BUF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_lo_tx_buf_en". This bit is ignored when SY_LO_TX_BUF_EN_OVRD_EN==0.
2 SY_LO_TX_BUF_EN_OVRD_EN	Override control for SY_LO_TX_BUF_EN 0b - Normal operation. 1b - Use the state of SY_LO_TX_BUF_EN_OVRD to override the signal "sy_lo_tx_buf_en".
1 SY_LO_RX_BUF_EN_OVRD	Override value for SY_LO_RX_BUF_EN When SY_LO_RX_BUF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_lo_rx_buf_en". This bit is ignored when SY_LO_RX_BUF_EN_OVRD_EN==0.
0 SY_LO_RX_BUF_EN_OVRD_EN	Override control for SY_LO_RX_BUF_EN 0b - Normal operation. 1b - Use the state of SY_LO_RX_BUF_EN_OVRD to override the signal "sy_lo_rx_buf_en".

45.3.5.3.1.73 TSM OVERRIDE REGISTER 2 (OVRD2)

45.3.5.3.1.73.1 Offset

Register	Offset
OVRD2	124h

45.3.5.3.1.73.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	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	RX_PHY_EN_OVRD	RX_PHY_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	TX_DIG_EN_OVRD	TX_DIG_EN_OVRD_EN
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	PLL_DIG_EN_OVRD	PLL_DIG_EN_OVRD_EN	RX_TZA_DCOC_EN_OVRD	RX_TZA_DCOC_EN_OVRD_EN	RX_TZA_PDET_EN_OVRD	RX_TZA_PDET_EN_OVRD_EN	RX_TZA_Q_EN_OVRD	RX_TZA_Q_EN_OVRD_EN	RX_TZA_I_EN_OVRD	RX_TZA_I_EN_OVRD_EN	RX_LNA_EN_OVRD	RX_LNA_EN_OVRD_EN	RX_BBA_DCOC_EN_OVRD	RX_BBA_DCOC_EN_OVRD_EN	RX_BBA_PDET_EN_OVRD	RX_BBA_PDET_EN_OVRD_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.5.3.1.73.3 Fields

Field	Function
31	Override value for FREQ_TARG_LD_EN

Table continues on the next page...

Field	Function
FREQ_TARG_LD_EN_OVRD	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.
30 FREQ_TARG_LD_EN_OVRD_EN	Override control for FREQ_TARG_LD_EN 0b - Normal operation. 1b - Use the state of FREQ_TARG_LD_EN_OVRD to override the signal "freq_targ_ld_en".
29 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.
28 DCOC_INIT_OVRD_EN	Override control for DCOC_INIT 0b - Normal operation. 1b - Use the state of DCOC_INIT_OVRD to override the signal "dcoc_init".
27 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.
26 DCOC_EN_OVRD_EN	Override control for DCOC_EN 0b - Normal operation. 1b - Use the state of DCOC_EN_OVRD to override the signal "dcoc_en".
25 RX_PHY_EN_OVRD	Override value for RX_PHY_EN When RX_PHY_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_phy_en". This bit is ignored when RX_PHY_EN_OVRD_EN==0.
24 RX_PHY_EN_OVRD_EN	Override control for RX_PHY_EN 0b - Normal operation. 1b - Use the state of RX_PHY_EN_OVRD to override the signal "rx_phy_en".
23 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.
22 SIGMA_DELTA_EN_OVRD_EN	Override control for SIGMA_DELTA_EN 0b - Normal operation. 1b - Use the state of SIGMA_DELTA_EN_OVRD to override the signal "sigma_delta_en".
21 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.
20 RX_INIT_OVRD_EN	Override control for RX_INIT 0b - Normal operation. 1b - Use the state of RX_INIT_OVRD to override the signal "rx_init".
19 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.
18 RX_DIG_EN_OVRD_EN	Override control for RX_DIG_EN 0b - Normal operation. 1b - Use the state of RX_DIG_EN_OVRD to override the signal "rx_dig_en".
17 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.

Table continues on the next page...

Field	Function
16 TX_DIG_EN_OVRD_EN	Override control for TX_DIG_EN 0b - Normal operation. 1b - Use the state of TX_DIG_EN_OVRD to override the signal "tx_dig_en".
15 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.
14 PLL_DIG_EN_OVRD_EN	Override control for PLL_DIG_EN 0b - Normal operation. 1b - Use the state of PLL_DIG_EN_OVRD to override the signal "pll_dig_en".
13 RX_TZA_DCOC_EN_OVRD	Override control for RX_TZA_DCOC_EN 0b - Normal operation. 1b - Use the state of RX_TZA_DCOC_EN_OVRD to override the signal "rx_tza_dcoc_en".
12 RX_TZA_DCOC_EN_OVRD_EN	Override control for RX_TZA_DCOC_EN 0b - Normal operation. 1b - Use the state of RX_TZA_DCOC_EN_OVRD to override the signal "rx_tza_dcoc_en".
11 RX_TZA_PDET_EN_OVRD	Override value for RX_TZA_PDET_EN When RX_TZA_PDET_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_tza_pdet_en". This bit is ignored when RX_TZA_PDET_EN_OVRD_EN==0.
10 RX_TZA_PDET_EN_OVRD_EN	Override control for RX_TZA_PDET_EN 0b - Normal operation. 1b - Use the state of RX_TZA_PDET_EN_OVRD to override the signal "rx_tza_pdet_en".
9 RX_TZA_Q_EN_OVRD	Override value for RX_TZA_Q_EN When RX_TZA_Q_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_tza_q_en". This bit is ignored when RX_TZA_Q_EN_OVRD_EN==0.
8 RX_TZA_Q_EN_OVRD_EN	Override control for RX_TZA_Q_EN 0b - Normal operation. 1b - Use the state of RX_TZA_Q_EN_OVRD to override the signal "rx_tza_q_en".
7 RX_TZA_I_EN_OVRD	Override value for RX_TZA_I_EN When RX_TZA_I_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_tza_i_en". This bit is ignored when RX_TZA_I_EN_OVRD_EN==0.
6 RX_TZA_I_EN_OVRD_EN	Override control for RX_TZA_I_EN 0b - Normal operation. 1b - Use the state of RX_TZA_I_EN_OVRD to override the signal "rx_tza_i_en".
5 RX_LNA_EN_OVRD	Override value for RX_LNA_EN When RX_LNA_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_lna_en". This bit is ignored when RX_LNA_EN_OVRD_EN==0.
4 RX_LNA_EN_OVRD_EN	Override control for RX_LNA_EN 0b - Normal operation. 1b - Use the state of RX_LNA_EN_OVRD to override the signal "rx_lna_en".
3 RX_BBA_DCOC_EN_OVRD	Override value for RX_BBA_DCOC_EN When RX_BBA_DCOC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_bba_dcoc_en". This bit is ignored when RX_BBA_DCOC_EN_OVRD_EN==0.
2	Override control for RX_BBA_DCOC_EN 0b - Normal operation.

Table continues on the next page...

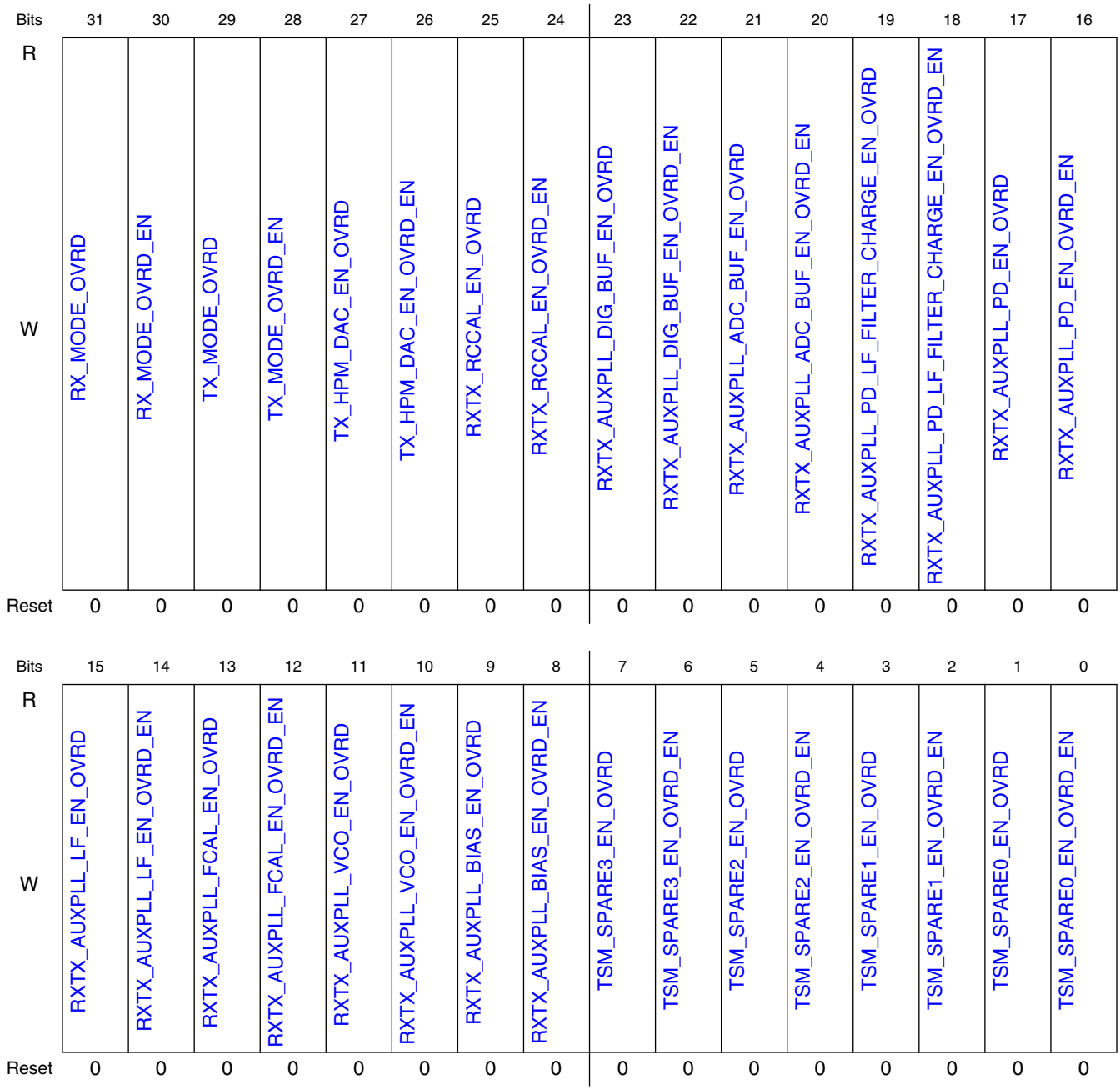
Field	Function
RX_BBA_DCOC_EN_OVRD_EN	1b - Use the state of RX_BBA_DCOC_EN_OVRD to override the signal "rx_bba_dcoc_en".
1 RX_BBA_PDET_EN_OVRD	Override value for RX_BBA_PDET_EN When RX_BBA_PDET_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_bba_pdet_en". This bit is ignored when RX_BBA_PDET_EN_OVRD_EN==0.
0 RX_BBA_PDET_EN_OVRD_EN	Override control for RX_BBA_PDET_EN 0b - Normal operation. 1b - Use the state of RX_BBA_PDET_EN_OVRD to override the signal "rx_bba_pdet_en".

45.3.5.3.1.74 TSM OVERRIDE REGISTER 3 (OVRD3)

45.3.5.3.1.74.1 Offset

Register	Offset
OVRD3	128h

45.3.5.3.1.74.2 Diagram



45.3.5.3.1.74.3 Fields

Field	Function
31 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.
30	Override control for RX_MODE

Table continues on the next page...

Field	Function
RX_MODE_OVRD_EN	0b - Normal operation. 1b - Use the state of RX_MODE_OVRD to override the signal "rx_mode".
29 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.
28 TX_MODE_OVRD_EN	Override control for TX_MODE 0b - Normal operation. 1b - Use the state of TX_MODE_OVRD to override the signal "tx_mode".
27 TX_HPM_DAC_EN_OVRD	Override value for TX_HPM_DAC_EN When TX_HPM_DAC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tx_hpm_dac_en". This bit is ignored when TX_HPM_DAC_EN_OVRD_EN==0.
26 TX_HPM_DAC_EN_OVRD_EN	Override control for TX_HPM_DAC_EN 0b - Normal operation. 1b - Use the state of TX_HPM_DAC_EN_OVRD to override the signal "tx_hpm_dac_en".
25 RXTX_RCCAL_EN_OVRD	Override value for RXTX_RCCAL_EN When RXTX_RCCAL_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_rccal_en". This bit is ignored when RXTX_RCCAL_EN_OVRD_EN==0.
24 RXTX_RCCAL_EN_OVRD_EN	Override control for RXTX_RCCAL_EN 0b - Normal operation. 1b - Use the state of RXTX_RCCAL_EN_OVRD to override the signal "rxtx_rccal_en".
23 RXTX_AUXPLL_DIG_BUF_EN_OVRD	Override value for RXTX_AUXPLL_DIG_BUF_EN When RXTX_AUXPLL_DIG_BUF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_dig_buf_en". This bit is ignored when RXTX_AUXPLL_DIG_BUF_EN_OVRD_EN==0.
22 RXTX_AUXPLL_DIG_BUF_EN_OVRD_EN	Override control for RXTX_AUXPLL_DIG_BUF_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_DIG_BUF_EN_OVRD to override the signal "rxtx_auxpll_dig_buf_en".
21 RXTX_AUXPLL_ADC_BUF_EN_OVRD	Override value for RXTX_AUXPLL_ADC_BUF_EN When RXTX_AUXPLL_ADC_BUF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_adc_buf_en". This bit is ignored when RXTX_AUXPLL_ADC_BUF_EN_OVRD_EN==0.
20 RXTX_AUXPLL_ADC_BUF_EN_OVRD_EN	Override control for RXTX_AUXPLL_ADC_BUF_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_ADC_BUF_EN_OVRD to override the signal "rxtx_auxpll_adc_buf_en".
19 RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_OVRD	Override value for RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN When RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_pd_lf_filter_charge_en". This bit is ignored when RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_OVRD_EN==0.
18 RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_OVRD_EN	Override control for RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_OVRD to override the signal "rxtx_auxpll_pd_lf_filter_charge_en".

Table continues on the next page...

Field	Function
17 RXTX_AUXPLL_PD_EN_OVRD	Override value for RXTX_AUXPLL_PD_EN When RXTX_AUXPLL_PD_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_pd_en". This bit is ignored when RXTX_AUXPLL_PD_EN_OVRD_EN==0.
16 RXTX_AUXPLL_PD_EN_OVRD_EN	Override control for RXTX_AUXPLL_PD_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_PD_EN_OVRD to override the signal "rxtx_auxpll_pd_en".
15 RXTX_AUXPLL_LF_EN_OVRD	Override value for RXTX_AUXPLL_LF_EN When RXTX_AUXPLL_LF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_lf_en". This bit is ignored when RXTX_AUXPLL_LF_EN_OVRD_EN==0.
14 RXTX_AUXPLL_LF_EN_OVRD_EN	Override control for RXTX_AUXPLL_LF_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_LF_EN_OVRD to override the signal "rxtx_auxpll_lf_en".
13 RXTX_AUXPLL_FCAL_EN_OVRD	Override value for RXTX_AUXPLL_FCAL_EN When RXTX_AUXPLL_FCAL_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_fcal_en". This bit is ignored when RXTX_AUXPLL_FCAL_EN_OVRD_EN==0.
12 RXTX_AUXPLL_FCAL_EN_OVRD_EN	Override control for RXTX_AUXPLL_FCAL_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_FCAL_EN_OVRD to override the signal "rxtx_auxpll_fcal_en".
11 RXTX_AUXPLL_VCO_EN_OVRD	Override value for RXTX_AUXPLL_VCO_EN When RXTX_AUXPLL_VCO_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_vco_en". This bit is ignored when RXTX_AUXPLL_VCO_EN_OVRD_EN==0.
10 RXTX_AUXPLL_VCO_EN_OVRD_EN	Override control for RXTX_AUXPLL_VCO_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_VCO_EN_OVRD to override the signal "rxtx_auxpll_vco_en".
9 RXTX_AUXPLL_BIAS_EN_OVRD	Override value for RXTX_AUXPLL_BIAS_EN When RXTX_AUXPLL_BIAS_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_bias_en". This bit is ignored when RXTX_AUXPLL_BIAS_EN_OVRD_EN==0.
8 RXTX_AUXPLL_BIAS_EN_OVRD_EN	Override control for RXTX_AUXPLL_BIAS_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_BIAS_EN_OVRD to override the signal "rxtx_auxpll_bias_en".
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 0b - Normal operation. 1b - Use the state of TSM_SPARE3_EN_OVRD to override the signal "tsm_spare3_en".
5	Override value for TSM_SPARE2_EN

Table continues on the next page...

Field	Function
TSM_SPARE2_EN_OVRD	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 0b - Normal operation. 1b - 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 0b - Normal operation. 1b - 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 0b - Normal operation. 1b - Use the state of TSM_SPARE0_EN_OVRD to override the signal "tsm_spare0_en".

45.3.6 CRCW

45.3.6.1 Introduction

The cyclic redundancy check and data whitening module (CRCW) contains logic necessary to provide CRC and Whitening for the radio link layer functions. Packet data is provided one bit at a time by the link layer functions on transmit, or by the radio physical layer on receive. Control and mode input signals dictate how the CRCW interprets and modifies the bit stream. The CRCW does not add any latency to the data as it is transmitted or received.

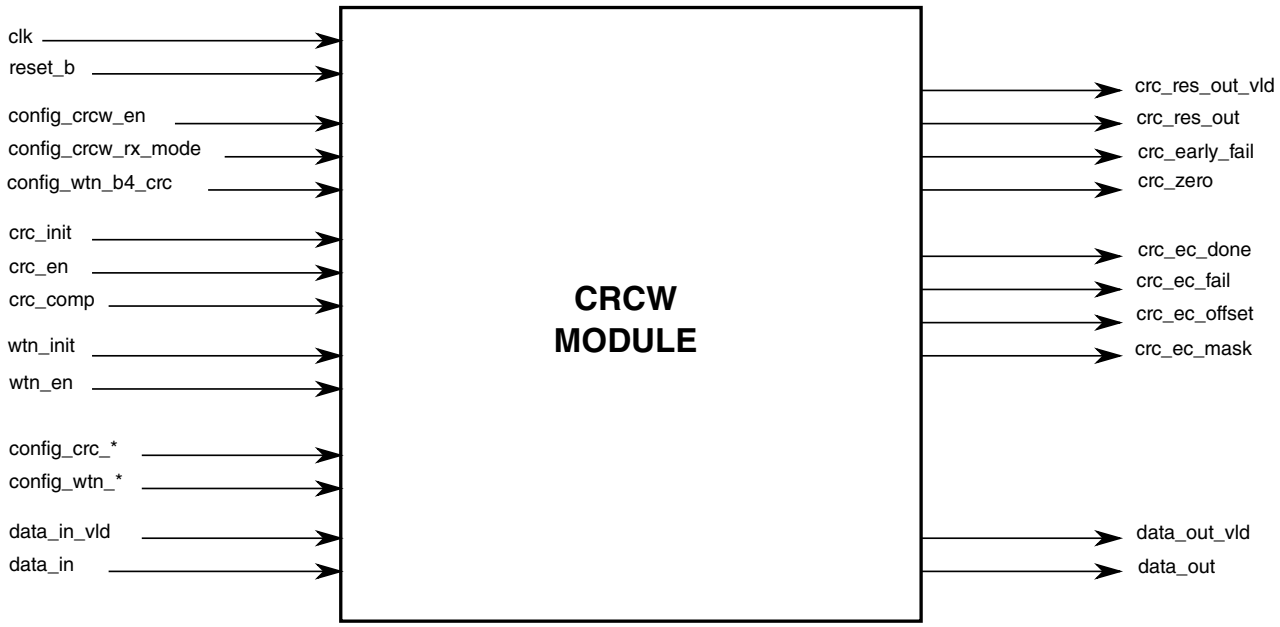


Figure 45-82. CRCW Block Diagram

45.3.6.1.1 Features

- Zero added data latency
- One module used for both transmit and receive
- All modes work at bit rates up to 1/10th the system clock rate
- Configurable CRC
  - 8/16/24/32-bit CRC register width
  - Programable polynomial and seed values
  - CRC value byte and bit order
  - Payoad input bit reflect
- Configurable Data Whitening
  - 1-bit to 9-bit LFSR width
  - Programable polynomial and seed values
  - Galois or Fibonacci type LFSR
  - Input data bit reflect
- Optional CRC Error Correction
  - Correct limited burst errors
  - Seperate settings for short and long packets
  - Offset/Mask reported to software for final data correction

45.3.6.1.2 Operation

The CRCW module contains the functionality to perform transmit CRC insertion, receive CRC checking, and data whitening. Additionally, the module can perform data error correction on received packets using the CRC syndrome if configured appropriately.

45.3.6.1.2.1 CRC Calculation

The CRC value is calculated using a configurable LFSR as depicted in Figure 45-83. The CRC calculation polynomial value is loaded into register *P*. The seed is loaded into register *SR* at initilization. The result is in register *SR* after all bits have been processed.

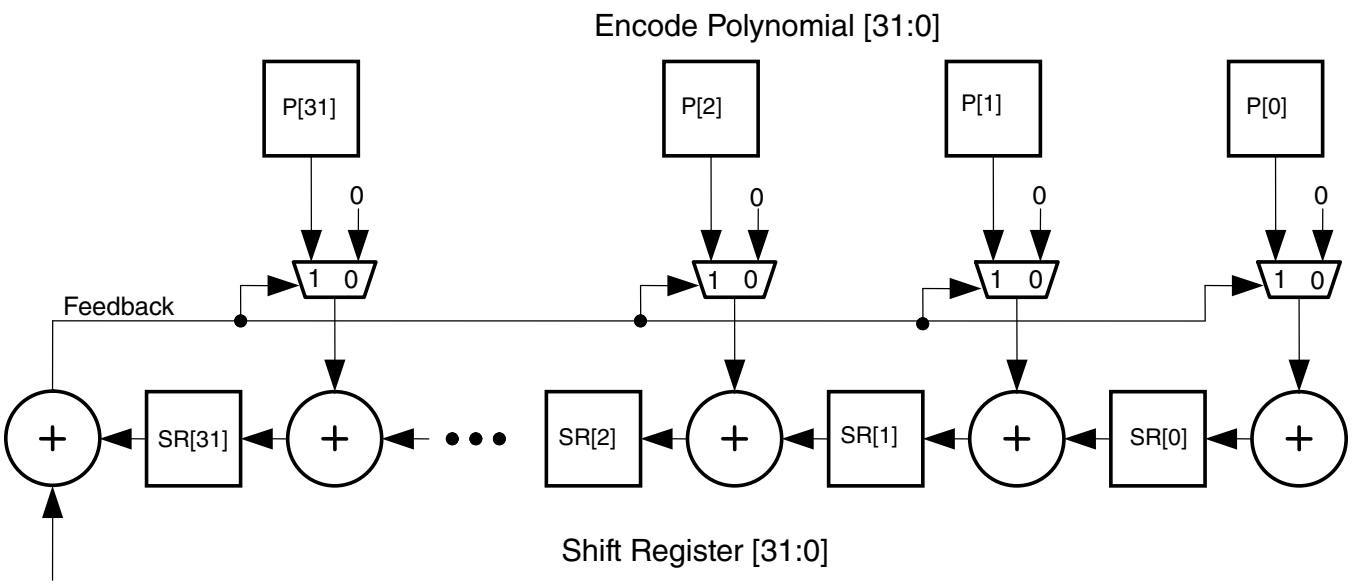
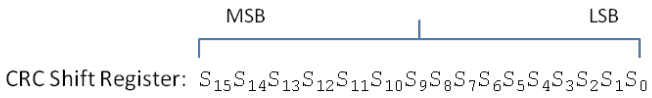


Figure 45-83. CRC Calculation Shift Register

Data flows from the link layer function to the CRCW module during transmit before being sent to the radio physical layer. All bits in each packet are provided by the link layer function, including placeholder bits for the CRC value. The link layer function marks the CRC calculation start and stop points in the bit stream. The link layer also indicates the start of the CRC placeholder bits. The CRCW module overwrites the placeholder CRC bits with the CRC result, ordered as specified by the configuration options. Figure 45-84 is a list of example 16-bit CRC output modes.



REF IN	BYTE ORDER	REF OUT	Order of Transmitted Bits (shifted out first ..... shifted out last)
X	0	0	$P_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}P_{14}P_{15}$ $S_{15}S_{14}S_{13}S_{12}S_{11}S_{10}S_9S_8S_7S_6S_5S_4S_3S_2S_1S_0$
X	0	1	$P_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}P_{14}P_{15}$ $S_8S_9S_{10}S_{11}S_{12}S_{13}S_{14}S_{15}S_0S_1S_2S_3S_4S_5S_6S_7$
X	1	0	$P_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}P_{14}P_{15}$ $S_7S_6S_5S_4S_3S_2S_1S_0S_{15}S_{14}S_{13}S_{12}S_{11}S_{10}S_9S_8$
X	1	1	$P_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}P_{14}P_{15}$ $S_0S_1S_2S_3S_4S_5S_6S_7S_8S_9S_{10}S_{11}S_{12}S_{13}S_{14}S_{15}$

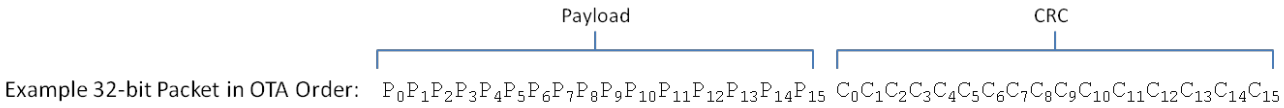
Figure 45-84. CRC Output Bit Order

45.3.6.1.2.2 CRC Check

During reception, data flows from the radio physical layer to the CRCW module before being sent to the link layer function. All bits in each packet are provided by the radio physical layer. Just as in transmit, the link layer function marks the CRC calculation start and stop points in the bit stream. The link layer also indicates the start of the received CRC bits.

The CRCW calculates the expected CRC result just as in transmit, but compares the received CRC bits as they arrive, ordered as specified by the configuration options, instead of inserting them into the packet. An early fail signal is asserted if a CRC bit miscompare is detected.

The CRC register will contain the expected CRC when either of the *config_crc_byte_order* or *config_crc_ref_in* CRC configuration options are set. The CRC register will contain the syndrome of the CRC error calculation when the both are not set.



REF IN	BYTE ORDER	REF OUT	Order of Bits Used in CRC Calculation (shifted in first ..... shifted in last)
0	0	0	$P_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}P_{14}P_{15}$ $C_0C_1C_2C_3C_4C_5C_6C_7C_8C_9C_{10}C_{11}C_{12}C_{13}C_{14}C_{15}$
0	0	1	$P_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}P_{14}P_{15}$ $C_7C_6C_5C_4C_3C_2C_1C_0C_{15}C_{14}C_{13}C_{12}C_{11}C_{10}C_9C_8$
0	1	0	$P_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}P_{14}P_{15}$ $C_8C_9C_{10}C_{11}C_{12}C_{13}C_{14}C_{15}C_0C_1C_2C_3C_4C_5C_6C_7$
0	1	1	$P_0P_1P_2P_3P_4P_5P_6P_7P_8P_9P_{10}P_{11}P_{12}P_{13}P_{14}P_{15}$ $C_{15}C_{14}C_{13}C_{12}C_{11}C_{10}C_9C_8C_7C_6C_5C_4C_3C_2C_1C_0$
1	0	0	$P_7P_6P_5P_4P_3P_2P_1P_0P_{15}P_{14}P_{13}P_{12}P_{11}P_{10}P_9P_8$ $C_0C_1C_2C_3C_4C_5C_6C_7C_8C_9C_{10}C_{11}C_{12}C_{13}C_{14}C_{15}$
1	0	1	$P_7P_6P_5P_4P_3P_2P_1P_0P_{15}P_{14}P_{13}P_{12}P_{11}P_{10}P_9P_8$ $C_7C_6C_5C_4C_3C_2C_1C_0C_{15}C_{14}C_{13}C_{12}C_{11}C_{10}C_9C_8$
1	1	0	$P_7P_6P_5P_4P_3P_2P_1P_0P_{15}P_{14}P_{13}P_{12}P_{11}P_{10}P_9P_8$ $C_8C_9C_{10}C_{11}C_{12}C_{13}C_{14}C_{15}C_0C_1C_2C_3C_4C_5C_6C_7$
1	1	1	$P_7P_6P_5P_4P_3P_2P_1P_0P_{15}P_{14}P_{13}P_{12}P_{11}P_{10}P_9P_8$ $C_{15}C_{14}C_{13}C_{12}C_{11}C_{10}C_9C_8C_7C_6C_5C_4C_3C_2C_1C_0$

Figure 45-85. CRC Input Bit Order

45.3.6.1.2.3 Data Whitening

The data whitener behaves identically in both transmit (whiten) and receive (de-whiten) operating modes - there is no distinction. The LFSR used in this operation is designed to use either Fibonacci or Galois types of calculations. In either case, the LSB of the shift register is used to whiten each data bit as they are streamed through the CRCW.

An initialization signal for the data whiten logic loads the shift register with the seed value. The initialization signal can be used to re-initialize the shift register during a packet. A protocol specific state machine controls a signal to mark the start and stop points for data whitening in the data stream.

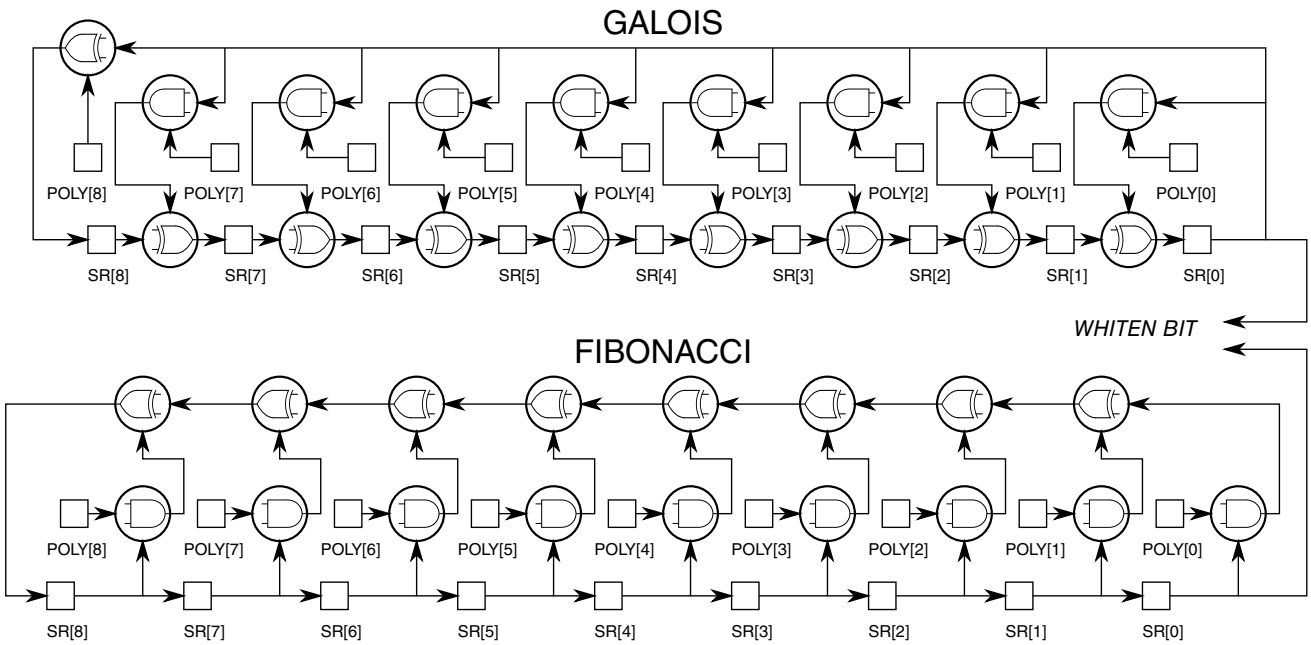


Figure 45-86. Data Whitening Shift Register

45.3.6.1.2.4 Error Correction

The CRC engine can optionally provide a level of error correction. This is accomplished by using the syndrome to locate the position of a possible burst error. This is especially useful for short burst errors. The error correction feature is controlled by configuration bits. A configuration setting defines the length of a short packet from zero to 31 bytes, and two different settings control the aggressiveness of error correction for short and long packets. More aggressive values attempt to correct longer burst errors, but potentially result in more false error correction solutions. Error correciton can be enabled for short packets and disasbled for long packets, or vice versa. The error correction feature is completely disabled when the CRC value is not received MSb first (*config_crc_byte_order* or *config_crc_ref_in*are enabled) because the error syndrome is not calculated.

While correcting random errors becomes difficult very quickly, it is fairly easy to correct a small burst error. Once the syndrome has been computed, all effects of the random message bits have been removed and the syndrome is the CRC value associated with the error signal. Reversing the CRC decoding process with the syndrome as the initial value of the LFSR will predict where the errors occurred in the payload. The structure of the CRC error correction LFSR is shown in Figure 45-87. The shifting process looks for a range of zeros in the least significant bits of the LFSR, counting the number of shifting operatations. The number of shifting operations determines the error location from the end of the packet accross all bits used in the CRC calculation.

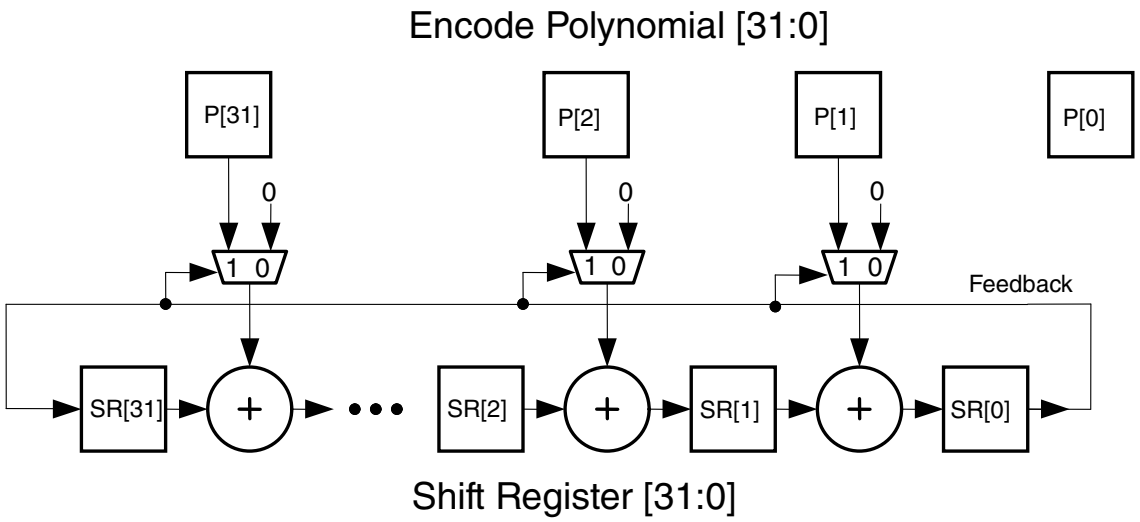


Figure 45-87. CRC Correction Shift Register

The CRCW module will perform the algorithm described above if enabled and the results of the CRC receive decode operation results in a syndrome that indicates an error. The logic performes one iteration per system clock cycle and can take up to *crc-included-bits* x *system clock cycles* to complete. The result of the error correction algorythm is an offset and XOR mask when successful. The offset is in bytes and counts only those bytes that are used in the CRC calculation, including the CRC value. The offset and mask values are zero on failure. The CRCW module will report if the algorythm was successful and the burst error position and mask so that upstream software can perform the correction.

45.3.6.2 Interface Timing

The CRCW interface timing requirements are described below. All signals except *reset_b* should be driven synchronous to *clk*. Some configuration options required a minumum number of clocks between data input bits or between the initialize input signals and the first data input bit.

The maximum input bitrate is $1/10^{th}$ the system clock rate as shown in Figure 45-88.

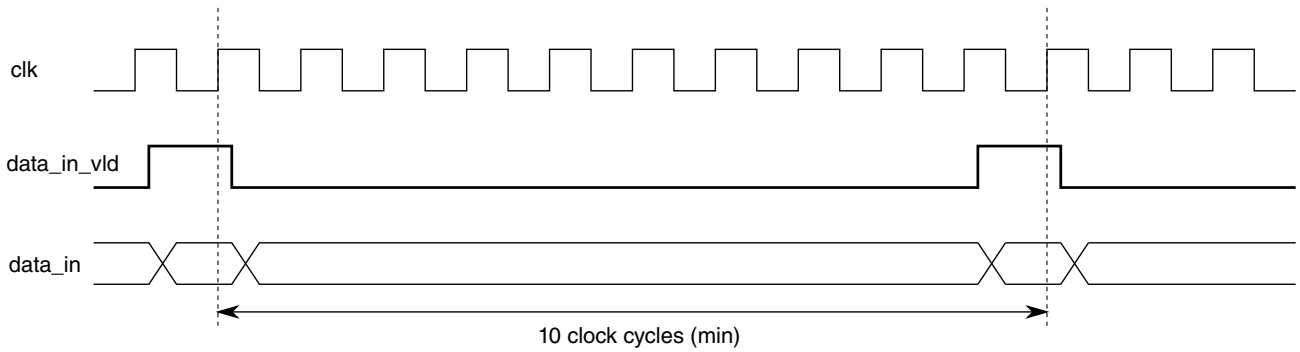


Figure 45-88. Maximum Bitrate Timing Diagram

45.3.6.2.1 Initialization

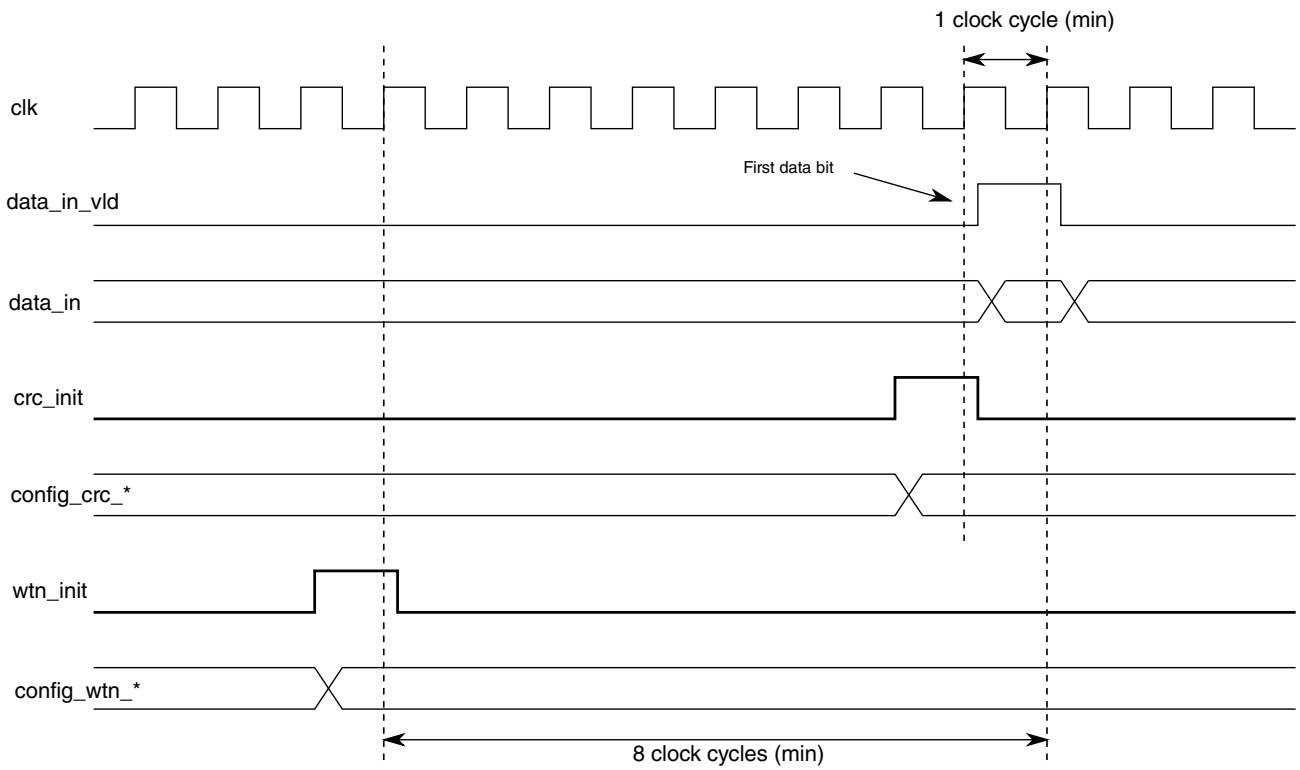


Figure 45-89. Initialization Timing Diagram

45.3.6.2.2 Whiten Re-Initialization

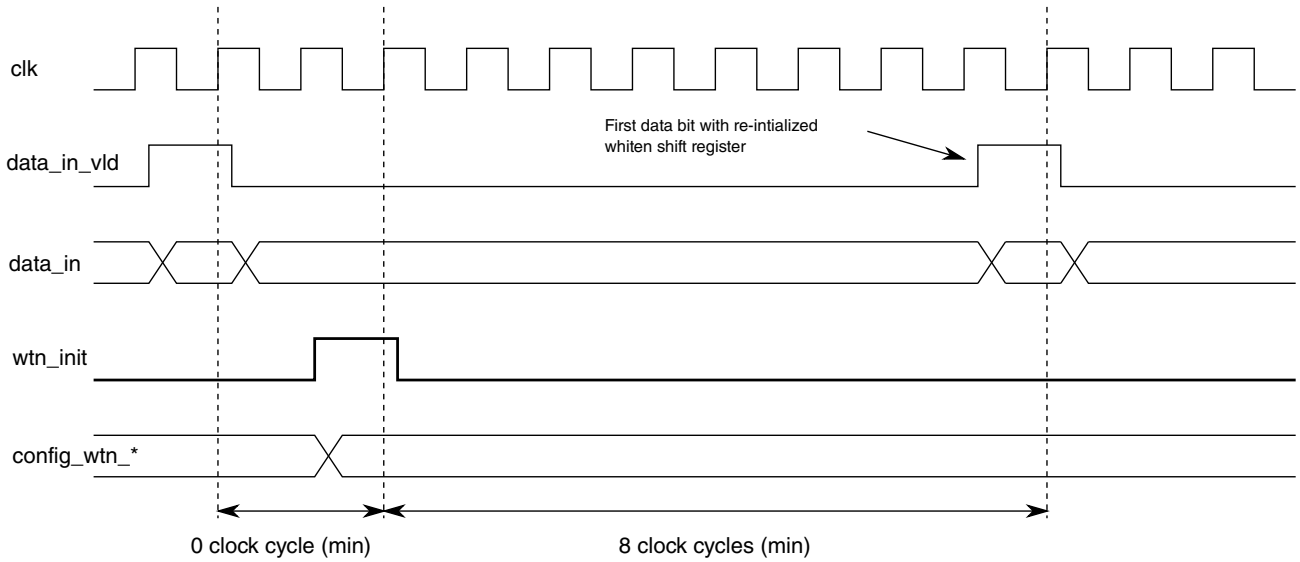


Figure 45-90. Whiten Re-Initialization Timing Diagram

45.3.6.2.3 CRC/WTN Enable

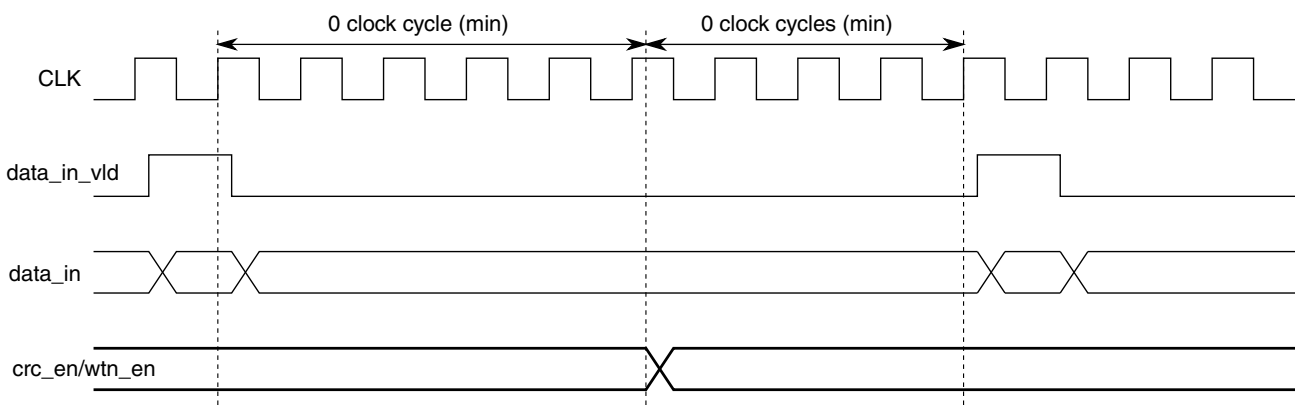


Figure 45-91. CRC/WTN Enable Timing Diagram

45.3.6.2.4 Results

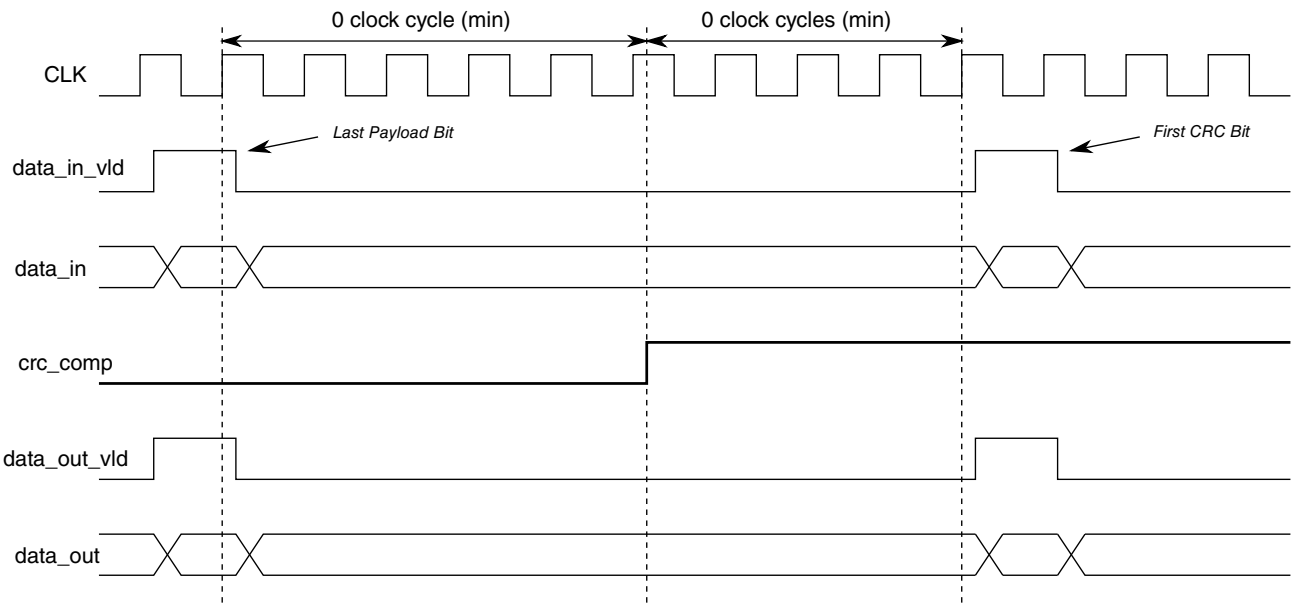


Figure 45-92. CRC Complete Timing Diagram

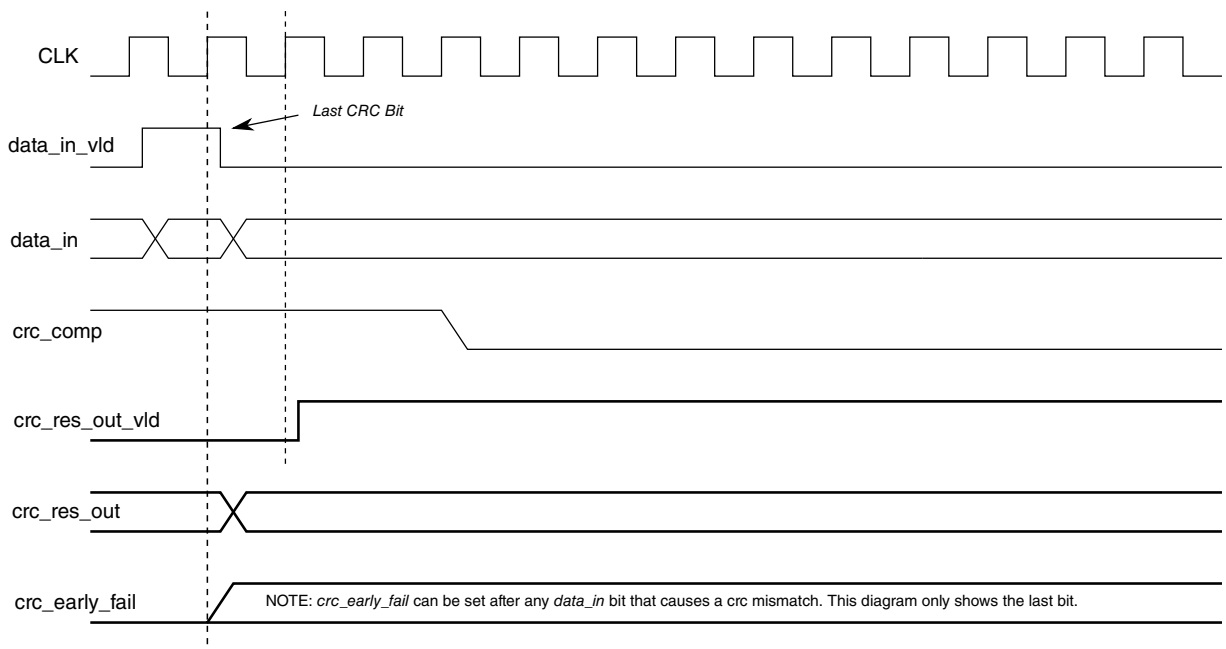


Figure 45-93. CRC Result Out Valid Timing Diagram

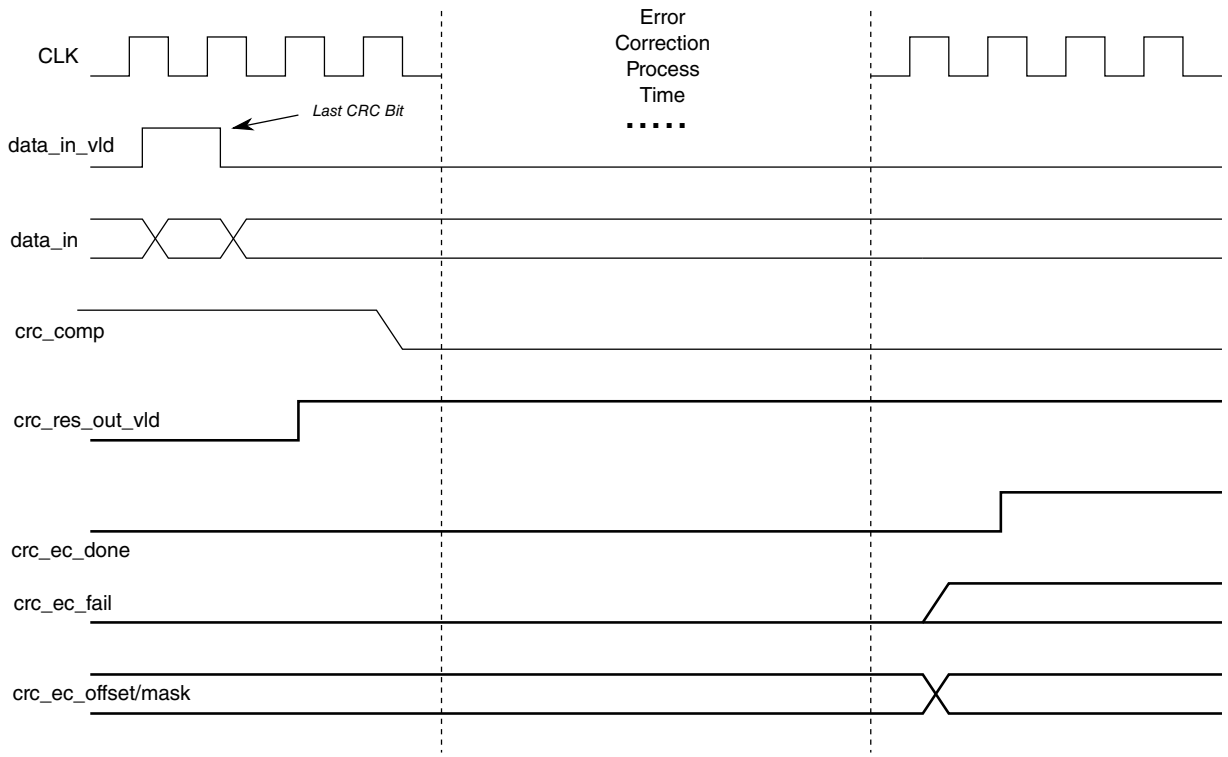


Figure 45-94. CRC Error Correction Done Timing Diagram

45.3.7 2.4 GHz PHY

45.3.7.1 Introduction

This documents describes the PHY and its operation. .

The Following protocols are supported:

- BLE
- custom FSK modulation schemes
- MSK

The PHY is designed such that it can support a variety of modulation schemes. Several parameters can be adjusted to suit each scheme as described in the configuration section.

45.3.7.1.1 Block diagram

The block diagram below illustrates the basic architecture of the PHY, but several details are not depicted for simplicity.

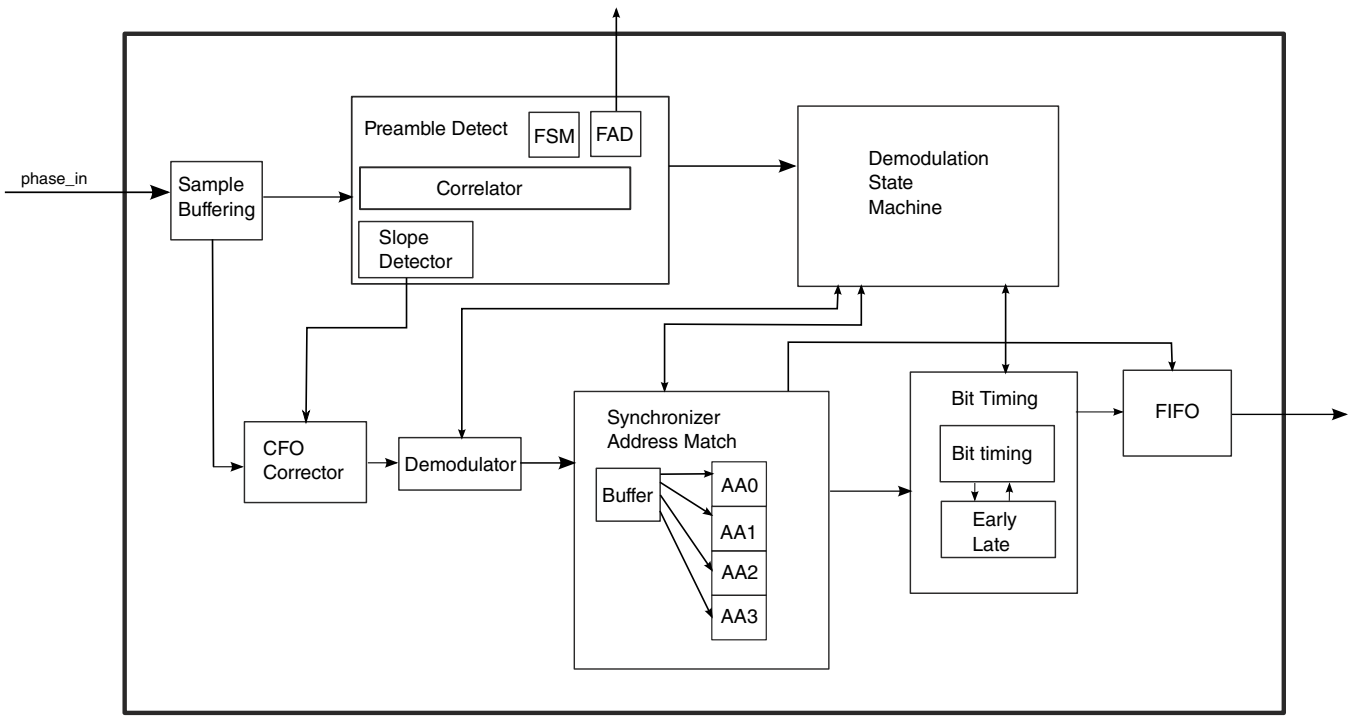


Figure 45-95. PHY Block diagram

45.3.7.1.2 Overview

The PHY is made up of several sub-modules. These modules can be viewed as members of two classes: datapath (the flow of the signal and the resultant bits) and control (enable/disable logic and state control).

45.3.7.1.2.1 PHY Datapath

The signal of interest is passed through the PHY along the data path. The overall objective is to process phase samples that are presented and create bits that are sent out. The significant sub-systems are here presented in generally chronological order as the signal passes through.

Initially, the phase samples are input, sequentially, at the sampling rate, one at a time. The first step is to buffer these samples, so they can be used all at once for certain computations. The buffer size is determined by the largest set of samples that will every be needed by any single component (48 samples).

The buffer of samples is then presented to the preamble detect (PD) logic. The PD module is responsible for detecting the presence of a signal and then indicating the timing and frequency offset when a signal is detected.

In parallel with the PD module, the CFO corrector is presented with the buffered samples. When a preamble is detected, the CFO estimate is presented to the corrector and the correction is applied to the incoming samples. In this fashion, the samples leaving the corrector will be optimally corrected for any frequency offset before beginning demodulation.

Demodulation is performed on frequency corrected samples after a preamble is detected. The produces soft information at the sample rate.

The soft information (soft bits) are then passed to the synchronization block. The primary objective of the synchronizer is to search the soft bits for a match to a sync pattern, typically referred to as an access address (AA). Up to 4 AA's can be searched for. When such a match occurs, the bit timing (the sample offset at which each bit is optimally demodulated) is recorded.

The bit timing block is then able to process the soft bits into hard bits by using the timing information from the synchronizer.

Finally the hard bits are prepared in a FIFO to allow an external block to read as desired. This is realized by a bit timing reference signal that times the output to match.

45.3.7.1.2.2 PHY Control

The control system of the PHY is largely composed of state machines. There are various direct control signals, but the most sophisticated logic is represented in three blocks: demod state machine, preamble detect state machine, and the FAD state machine.

The demodulation state machine has the master control over the entire PHY. When activated, it initiates the preamble search and decides when to activate the demodulator and bit timing engines.

The preamble detect state machine controls the correlator logic and makes the decision when a preamble has been found. Different search mechanisms are used based on the PD_MODE selected.

The fast antenna diversity (FAD) state machine is essentially a replacement of the preamble detect state machine when FAD is enabled. It has the task of detecting the preamble and deciding which antenna to stay on for further processing.

45.3.7.2 Interfaces

The PHY interacts with the system in a few distinct ways. The interfaces can be grouped into 4 categories: configuration, signal input, demodulation output, and miscellaneous.

PHY Configuration Interface

The PHY is configured by static signals that are observed and used throughout the receive process. It is important that these signals (typically originating from IP registers) remain static during a receive operation to avoid unpredictable behavior. Some examples of these types of configurations are PD thresholds, AA to search for, etc.

Signal Input Interface

The PHY processes the inputs signal from a signed, 5-bit phase inputs. This input is expected to be marked as 'valid' on the phase_in_valid signal. The phase_in_valid signal must be synchronous with input 'clk', but should only assert for 1 clock cycle when a new sample is presented. It is also required that at lease 3 clocks (clk) be present per each phase_in_valid. This is to ensure that all pipe-lined operations can complete.

Demodulation Output

The demodulator produces bits and, after re-timing, sends them out. The bits are re-timed to the input bit_timing_en which is assumed to be the system bit clock. It is required that the bit_timing_en is synchronous with the input clock and phase_in_valid signals, as synchronization is not performed in the PHY, only re-timing.

Miscellaneous

There are various other outputs from the PHY that are mostly intended for debug. A few significant signals are described here:

- Soft Info (soft bits): A representation of the demodulator output in fixed-point format. These outputs can be used to see how strong the bits are and the inter-symbol behavior.
- CFO: Frequency offset as determined by the preamble detect engine.
- corr_mag: correlator magnitude, a representation of the correlation strength between the input signal and the reference preamble.
- aa_matched: indicates that an acceptable address synchronization has occurred.
- ant_sw: indicates the requested antenna when FAD is enabled. May be ignored when FAD is not enabled.

45.3.7.3 Functional Description

This section describes the functional operation of the PHY.

45.3.7.3.1 Protocol Configuration

The PHY has several configurable parameters that can be changed based on the desired protocol behavior. The details of these parameters are described in the following sections and a summary table is provided here for quick reference.

Table 45-36. FSK schemes supported by the demodulator

Scheme	R baud rate (kbps)	*h* modulation index	Frequency pulse	Peak LO error (kHz)
BLE	1000	0.50	Gaussian; BT=0.5	±300
ANT	1000	0.32	Gaussian; BT=0.5	±200
Custom scheme 1	500	0.70	Gaussian, BT=0.5	±200
Custom scheme 2	1000	0.50	Gaussian, BT=0.3	±200
Custom scheme 3	500	0.50	Half-sine pulse, L=1	±200
Custom scheme 4	250	0.50	Gaussian, BT=0.5	±100
Custom scheme 5	250	1.00	Gaussian, BT=0.5	±100
Custom scheme 6	1000	0.50	Gaussian, BT=0.7	±300
Custom scheme 7	2000	0.50	Gaussian, BT=0.5	±300
Custom scheme 8	250	0.50	Half-sine pulse, L=1	±100
Custom scheme 9	2000	0.30	Gaussian, BT=0.5	±200
Custom scheme 10	500	0.50	Rectangular pulse, L=1	±200

Table 45-37. PHY Configurations

Config	PD Mode Normal	PD Thresh	PD Mode Comp.	PD Thresh Compressed	AA Corr Gain	PD FREQ THRESH	Preamble Reference {PRE_REF7, PRE_REF6, PRE_REF5, PRE_REF4, PRE_REF3, PRE_REF2, PRE_REF1, PRE_REF0}
BLE	2	0.8 = 0xCD	2	0.9 = 0xE6	0.875 = 0x07	0x0A	{0x1e,0x1e,0x00,0x02,0x02,0x02,0x00,0x1e}
ANT	3	0.9 = 0xE6	2	0.9375 = 0xF0	1.5 = 0x0C	0x06	{0x1e,0x1f,0x00,0x01,0x02,0x01,0x00,0x1f}
CS1	2	0.675 = 0xAD	2	0.9 = 0xE6	0.75 = 0x06	0x0D	{0x1d,0x1e,0x00,0x02,0x03,0x02,0x00,0x1e}
CS2	3	0.9 = 0xE6	3	0.93 = 0xEE	1.0 = 0x08	0x06	{0x1f,0x1f,0x00,0x01,0x01,0x01,0x00,0x1f}
CS3	2	0.75 = 0xC0	2	0.9 = 0xE6	0.875 = 0x07	0x0D	{0x1d,0x1e,0x00,0x02,0x03,0x02,0x00,0x1e}
CS4	2	0.8 = 0xCD	2	0.9 = 0xE6	0.875 = 0x07	0x0D	{0x1e,0x1e,0x00,0x02,0x02,0x02,0x00,0x1e}
CS5	2	0.6875 = 0xB0	2	0.8 = 0xCD	0.875 = 0x07	0x0D	{0x1b,0x1d,0x00,0x03,0x05,0x03,0x00,0x1d}
CS6	2	0.85 = 0xD9	2	0.9 = 0xE6	0.875 = 0x07	0x0A	{0x1d,0x1e,0x00,0x02,0x03,0x02,0x00,0x1e}
CS7	3	0.8 = 0xCD	2	0.93 = 0xEE	1.25 = 0x0A	0x05	{0x1e,0x1e,0x00,0x02,0x02,0x02,0x00,0x1e}

Table continues on the next page...

Table 45-37. PHY Configurations (continued)

CS8	2	0.75 = 0xC0	2	0.9 = 0xE6	1.0 = 0x08	0x0D	{0x1d,0x1e,0x00,0x02,0x03,0x02,0x00,0x1e}
CS9	3	0.9 = 0xE6	3	0.93 = 0xEE	2.0 = 0x10	0x03	{0x1f,0x1f,0x00,0x01,0x01,0x01,0x00,0x1f}
CS10	2	0.82 = 0xD3	2	0.9 = 0xE6	1.0 = 0x08	0x0D	{0x1e,0x1e,0x1f,0x01,0x02,0x02,0x01,0x1f}

45.3.7.3.2 Scan and Preamble Detect

The initial operation of the PHY when it is enabled is in a SCAN state. The purpose of this operation is to scan the input samples for a preamble, so this state is alternately referred to as 'Preamble Detect' or 'PD'. Additionally, when a preamble is detected, the PHY acquires a coarse time reference for the incoming packet. So this process is also sometimes referred to as coarse timing search (CTS).

45.3.7.3.2.1 Correlation

To accomplish preamble detection, a correlator is used to assess the similarity of the incoming phase samples with a reference waveform. The reference waveform is 6 symbols long. The PHY is configured to look for a particular waveform by programming phase values into the PRE_REF*n* IP register fields. However, there are only 8 fields. These represent 2 symbols of 4 samples each. Internally, the PHY replicates this reference to produce a 6 symbol reference. Therefore, it is required that the preamble be repetitive on a 2 symbol basis.

As the phase samples are received by the PHY, they are buffered. The buffer depth is 48 samples so it contains the last 12 potential symbols. From this buffer, samples are selected to compute a frequency offset (CFO) that may be present because of crystal inaccuracy, etc. The number of symbols over which the frequency estimate is computed is determined by the configuration of the PHY. For normal, 6-symbol preamble detect, the frequency estimate is calculated based on the frequency difference over 6 symbols (24 samples). For compressed, 4-symbol mode, the estimate is over 4 symbols (16 samples). For FAD, the estimate is computed over 2 or 4 symbols, depending on the current state of the search. This estimate is continually updated during a preamble search. It is latched when a preamble is declared found.

For clarity, normal, 6-symbol correlation is described, but the other modes operate similarly. As the CFO is calculated, it is applied immediately over the last 24 samples. This is done by generating a phase ramp that would be caused by the frequency offset and then subtracting it from the last 24 samples. This produces a frequency corrected buffer of 24 samples.

The reference waveform is then subtracted from the frequency corrected samples. These differences are converted to the I/Q domain using a look-up table so that a magnitude can be computed. The correlation magnitude is the result of calculating the I/Q mean over the previous 6 symbols and then squaring. The resultant value is referred to as the correlation magnitude or correlation value. It serves as an indicator of how well the previous 6 symbols correlate with the desired reference preamble and updates on every sample.

As previously mentioned, there are 2 distinct modes for the correlator, compressed and normal. The difference is that compressed mode only correlates based on 4 symbols.

45.3.7.3.2.2 Detection

The correlation magnitude provides a measure of how well correlated the incoming signal is to the reference waveform. So, the higher the magnitude, the more likely there is a real preamble present. The magnitude is compared to a threshold that depends on the given protocol and configuration of compressed mode. For every sample that the magnitude is greater than the threshold, a 'pass' event is created. The pass events are processed differently depending on which PD mode is configured.

PD mode 3 is a detection mode that simply looks for the correlation magnitude to exceed the threshold. If it does, then another check is performed 2 symbols (8 samples) later to confirm the finding. If it is confirmed, a preamble is declared 'found' and the CFO estimate is locked for the demodulator.

PD mode 2 also searches for magnitudes that exceed the programmed threshold. However, is also looks for signature peaks. This means that a preamble is only declared 'found' if 3 peaks, above the threshold, and 2 associated 'dips' between the peaks, are observed. This peak and dip behavior is a result of the preamble being negatively correlated when shifted by one symbol.

In either case, when a preamble is declared 'found', a signal is sent to the rest of the PHY that the CFO is to be locked and demodulation can begin for address search and synchronization.

It is important to not that until an address is found, preamble search continues. This can allow for a refinement in the CFO value after initial declaration of a found preamble.

45.3.7.3.2.3 FAD

A simple fast antenna diversity (FAD) mode is provided to allow for 2 antenna scanning. If FAD mode is enabled, the incoming stream of samples is scanned for a preamble just as in the normal mode, but on a significantly reduced scale. The initial operation of FAD checks for correlation over 3 symbols, using a CFO estimate calculated over 2 symbols.

The FAD search switching is configured by setting the FAD_PROC_DUR and FAD_PROC_DLY parameters. The delay parameter allows for antenna switch time and processing delay of the samples. The duration parameter determines the time at which the antenna should switch.

The antenna switch output will continue to toggle until a viable preamble detection occurs. When it does, the antenna will stay in the same state to allow for demodulation. Additionally, regular preamble search is resumed so a refined CFO estimate is possible.

45.3.7.3.3 Demodulation

As the PD logic detects a preamble, the CFO estimate is locked and this estimate is used to correct all phase samples before entering the demodulator. The overall purpose of the demodulator is to process the frequency corrected phase samples and produce soft-bit values at each sample time. This results in a soft-bit stream at 4x the bit/symbol rate.

The corrected phase samples are passed into the demodulator and then buffered. The buffer size is 8, so 2 symbols will fit. The purpose of having 2 symbols is so that a probabilistic estimate can be used to predict the desired bit level.

The output of the demodulator is a signed sfix10n8 soft representation of the demodulated bit. The soft bits are output at the sample rate, so there are 4 soft values per bit. This allows for fractional timing and early/late adjustment elsewhere in the PHY.

45.3.7.3.4 Address Search and Synchronization

When a preamble has been detected, the synchronization block is activated. The purpose of this block is to analyze the demodulated soft bits and look for an Access Address (AA). The PHY supports up to 4 AA's to be searched for and they can vary in size from 1 to 4 octets. When an address is detected, the system is notified, a fine timing estimate of the symbol boundary is produced, and hard bit decisions begin. For each AA to be searched for, an independent engine is enabled so that all 4 can potentially be searched in parallel.

As the soft bits enter the synchronization engine, they are buffered after being multiplied by AA_CORR_GAIN. This gain is protocol dependent to properly scale the soft bits for ideal detection. The buffer size is 128 to allow for the maximum size address to fit (32 bits x 4 samples/bit). The representation of the soft bits is reduced to sfix4n2 in the buffer.

The buffered soft bits are then compared in two ways to the desired AA: hard and soft.

For hard AA detection, the sign bit of the soft bit is used and a hamming distance is calculated using the programmed AA. If the calculated hamming distance is less than or equal to the programmed allowable hamming distance for the particular address, a hard AA match is indicated.

For soft detection, the soft bits are multiplied by the associated desired bit (as a sign) and then all the products are summed. The result is a correlated strength indication.

For a successful AA match to occur, the hamming distance must be met and there must be a peak in the soft correlation. The location of the peak determine the exact position of the AA and the fractional timing.

At this point the system is notified of an AA match and the synchronization machine is shut off.

AA confirmation

The synchronization block also contains a special engine that is used to turn off the preamble detector under certain conditions. If enabled, the AA confirm block will look at the soft bits for a sequence of bits that indicates that the AA has begun and will signal the PD engine to shut off. The purpose of this is to reduce the risk of further false preamble detections that may cause an erroneous frequency estimate to be stored. The 6-bit pattern that is searched for depends on where in the AA the first doublet occurs. For example, if the expected bit stream (including the preamble) is [0 1 0 1 0 1 0 1 0 0 1 0 1 1 ....], then the pattern to be confirmed is [0 1 0 1 0 0]. This feature is only available for AA0.

45.3.7.3.5 Bit Timing and Production

After an address is synchronized, the PHY begins to produce hard bits, using the demodulated soft bits and the timing information obtained during synchronization.

45.3.7.3.5.1 Bit Production

Bits are produced nominally based on the sign of the soft bit, at the proper sample time. However, for enhanced performance, the PHY performs fractional adjustment to optimally decode the bit. The fractional timing initially provided by the synchronization engine is used to combine soft bits before and after the desired sample time. A weighted summation is performed based on this fraction and then the sign of the result is inverted to produce the bit. Along with the bit, an internal bit valid strobe is generated at one-fourth the sample rate to indicate a valid bit.

45.3.7.3.5.2 Early/Late

The early/late (EL) engine allows for the bit timing module to adjust the timing of the eye of each bit to account for drift. When the EL engine detects that the eye of the bit has moved in time, an adjustment signal is generated and the bit timing is accordingly adjusted. The EL engine is configured for the window over which an EL measurement is taken (EL_WINDOW) and the duration for which each timing adjustment is considered valid (EL_INTERVAL).

45.3.7.3.5.3 Bit Synchronization

Because the relationship between the sample clock and the bit eye is not knowable *a priori*, it is necessary to buffer the bits and then re-time them to the system bit clock. The PHY's version of the bit clock is guaranteed to be synchronous to the system clock as both are divided down from the same reference clock. However the alignment is not certain. Therefore a 16 entry FIFO holds bits as generated by the PHY and then outputs them at the clock edges indicated by the system.

45.3.7.3.5.4 Special Case for Bit Processing: MSK

When MSK mode is enabled there are a couple special processing steps that take place. The desired AA that is programmed is first transformed to an equivalent bit pattern as it would be observed over the air using the FSK demodulator. This allows the synchronizer to properly search for the incoming address. Additionally, the final bit of the address is stored so that it may be used for the bit stream processing.

When a packet is detected in MSK mode, the bits are demodulated normally, then, starting with the last bit of the AA, they are transformed serially into a true MSK bit stream to account for the transformation that is caused by using the FSK demodulator. The actual bits that are presented out of the PHY are, therefore, the expected bits and no further transform should be required.

45.3.8 Transceiver DMA and Packet RAM Debug Modes

45.3.8.1 Introduction

For debug and evaluation purposes, the 2.4GHz Radio provides the capability to capture a variety of internal data during the reception process, and store it to either system memory (using Transceiver DMA mode) or radio Packet RAM (using Packet RAM Debug mode), for analysis and post-processing

45.3.8.1.1 Overview

During silicon debug and evaluation, it is often desirable to gain access to data internal to the digital receiver, in order to, for example:

- troubleshoot packet loss issues
- troubleshoot unexpectedly high Bit Error Rate
- fine-tune receiver initialization and configuration parameters

The 2.4GHz radio offers 3 methods of accessing such internal data:

1. Transceiver DMA mode
2. Packet RAM Debug mode
3. Digital Teset Mux (DTEST)

This chapter describes the first 2 methods.

Examples of internal data desirable for capture are raw ADC samples, RX digital I and Q outputs, and PHY soft decision data. Transceiver DMA and Packet RAM Debug modes allow a source of internal data to be selected, automate the acquisition of the data, pack the data into a format suitable and optimized for its destination (system or radio memory), implement the interfaces required to access those memories, store the data to memory under program control, and subsequently allow host MCU access to the acquired data for analysis and post-processing. Selected data are organized into pages, to allow contextually related data to be simultaneously captured wherever possible, and allow for maximum utilization of available bandwidth to memory. For DMA mode, two transfer protocols are provided: a low-bandwidth protocol where the radio requests access to memory only when it has a new sample ready, and a high-bandwidth protocol that allows an entire block of radio data to be transferred to memory with a single radio request. For RAM debug mode, the entire radio RAM is made available for this debug mode, with hardware to pack data, generate RAM addressing, and detect RAM-full condition to prevent overwriting and loss of data.

45.3.8.1.2 Features

- Captures any of 4 sources of RX_DIG data and 4 sources of PHY data
- DMA engine communicates with SoC DMA2 controller to access system memory
- DMA signalling compliant with IP Interface SRS DarkBlue Line convention
- DMA engine provides 2 options to optimize request signalling to required data rate
- Packet RAM Debug engine automates transfers to radio memory
- Efficient packing of source data optimized to destination memory dimensions
- Efficient packing of source data optimized to make best use of available bandwidth
- DMA and Packet RAM engine clocking disabled in mission mode to reduce power
- Simultaneous DMA and Packet RAM Debug operation possible

- Simultaneous Mission Mode and DMA operation possible
- Hardware start-triggering capability with 7 hardware sources
- Stop-on-trigger capability for Packet RAM Debug with 11 hardware sources

45.3.8.1.3 Block Diagram

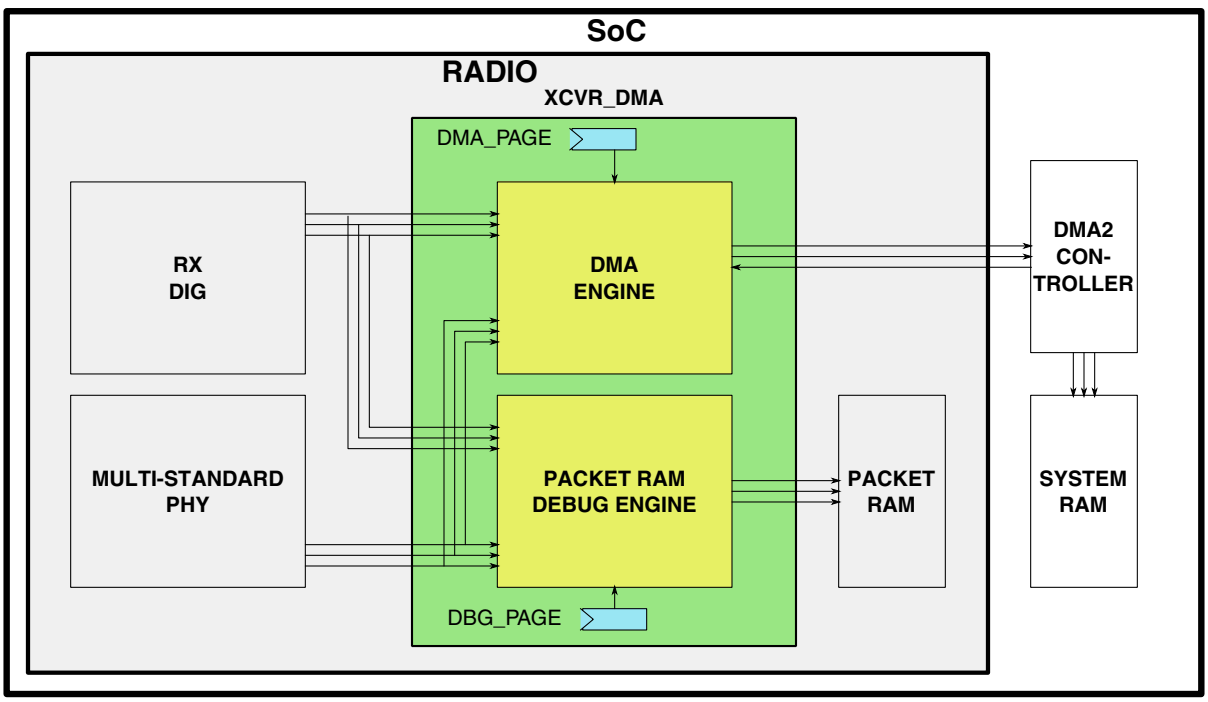


Figure 45-96. Transceiver DMA and Packet RAM Debug

45.3.8.2 Memory Map and Register Definition

45.3.8.2.1 Register Summary

Below is an overview of the DMA and Packet RAM Debug control fields. More detail on these registers can be found in the XCVR_MISC register section.

45.3.8.2.2 Register Descriptions

Field	R/W	Description
DMA_PAGE[3:0]	rw	Transceiver DMA Page Selector. Selects the page of receiver data for storage to system memory when using Transceiver DMA Debug Mode. Setting this register to a non-zero value enables the DMA interface logic. Setting this register to zero disables the interface. The available DMA pages are listed in the following Table 45-37.
DMA_EN	rw	Transceiver DMA Enable. Setting DMA_EN=1 enables clock gating to the Transciever DMA engine. This bit should be set prior to the start of a DMA session, and remain set for the duration of the session.
SINGLE_REQ_MODE	rw	DMA Single Request Mode. Configures the transceiver to transfer the entire block of DMA data with only a single initial request (ipd_req_radio_rx). The DMA2 controller must be configured accordingly. In this mode, the transceiver will use **ips_xfr_wait** to pace the individual transactions. Single Request Mode should not be used with DMA Pages 11, 12, or 13, because the data rate is too low and therefore ips_xfr_wait would remain assserted for excessively long periods.
BYPASS_DMA_SYNC	rw	Bypass External DMA Synchronization. When using transceiver DMA with SINGLE_REQ_MODE=0, synchronization external to the transceiver must be bypassed in order to minimize bus access latency. Using DMA in this mode also requires that the MCU and transceiver both are configured to operate in the RF Oscillator clock domain.
DMA_AA_TRIGGERED	r	DMA AA_TRIGGERED. This status bit becomes set when the PHY indicates that a Network Address match has occurred. In some cases it may be desirable to delay the start of DMA transfers until such a match occurs; in such cases, software can poll this bit, and once it becomes set, immediately select the desired DMA_PAGE to initiate DMA transfers. This bit is intended for use with DMA_PAGE=13, but is available on all pages. Its usage is optional.
DMA_TIMED_OUT	r	DMA Transfer Timed Out. Status bit indicates that the transceiver DMA, while operating in Single Request Mode, asserted ips_xfr_wait for a period in excess of the programmed **DMA_TIMEOUT** setting, resulting in a timeout condition where the transceiver has forced ips_xfr_wait low. After a timeout, before resuming DMA operations, set DMA_PAGE=0 to disable DMA, and write 1 to this bit to clear it.
DMA_TIMEOUT[3:0]	rw	DMA Timeout. In DMA Single Request Mode, adverse consequences may result if the transceiver's ips_xfr_wait signal is asserted for an excessively long period of time, which could occur due to mis-programming of the DMA2 controller, and/or the transceiver itself. DMA Timeout forces the transceiver's ips_xfr_wait low after *N* microseconds and sets the **DMA_TIMED_OUT** status bit, where *N*=DMA_TIMEOUT. DMA_TIMEOUT is only relavent when SINGLE_REQ_MODE=1, otherwise the transceiver does not assert ips_xfr_wait.
DMA_START_TRG[2:0]	rw	The DMA Start Trigger, if desired, can be selected from the sources in the following table: 0: no trigger 1: FSK PHY: gfsk_preamble_found 2: FSK PHY: aa_sfd_matched

Table continues on the next page...

Field	R/W	Description
		3: Reserved 4: Reserved 5: RXDIG: agc_dcoc_gain_chg 6: TSM: rx_dig_en 7: TSM: tsm_spare2_en
DMA_START_EDGE	rw	DMA_START_EDGE selects the edge sensitivity (rising or falling) of the selected start-trigger. 0: Trigger fires on a rising edge of the selected trigger source 1: Trigger fires on a falling edge of the selected trigger source
DMA_START_TRIGGERED	r	DMA Start Trigger Occurred. This read-only status bit becomes set, when, during a DMA session (DMA_PAGE > 0), the trigger source selected by DMA_START_TRG[2:0] occurs, with the edge sensitivity selected by DMA_START_EDGE. To clear this bit, set DMA_PAGE=0.
DBG_PAGE[3:0]	rw	Packet RAM Debug Page Selector. Selects the page of receiver data for storage to Packet RAM using Transceiver Packet RAM Debug Mode. Setting this register to a non-zero value enables the Packet RAM Debug mode interface logic. Setting this register to zero disables the interface. The available RAM Debug pages are listed in the following Table 45-37
DBG_EN	rw	Packet RAM Debug Mode Enable. Setting DBG_EN=1 enables clock gating to the Packet RAM Debug engine. This bit should be set prior to the start of a Debug session, and remain set for the duration of the session.
XCVR_RAM_ALLOW	rw	Allow Packet RAM Transceiver Access. This bit must be set before performing accesses to the Packet RAM using transceiver address space. Transceiver space accesses to Packet RAM are intended for debug purposes only; the individual protocol engines, and the associated IPS busses, have exclusive access to Packet RAM in mission modes. Transceiver space accesses to Packet RAM include direct accesses to RAM at transceiver addresses 0x700 - 0xFFF, as well as Packet RAM Debug Mode. When this bit is set, control of RAM clock gating and RAM chip enables are forced on continuously, taking these controls away from the protocol engines. **Note:** In Packet RAM Debug mode, this bit should be set to 1 first, prior to setting DBG_PAGE to any non-zero setting
XCVR_RAM_PAGE	rw	This bit selects which of the 2 Packet RAM blocks is mapped into XCVR address space starting at XCVR_BASE + 0x700. 0: RAM0 is mapped into XCVR address space, between XCVR_BASE + 0x700, and XCVR_BASE + 0xFFF 1: RAM1 is mapped into XCVR address space, between XCVR_BASE + 0x700, and XCVR_BASE + 0xFFF
DBG_AA_TRIGGERED	r	DBG_AA_TRIGGERED. This status bit becomes set when the PHY indicates that a Network Address match has occurred. In some cases it may be desirable to delay the start of Packet RAM debug transfers until such a match occurs; in such cases, software can poll this bit, and once it becomes set, immediately select the desired DBG_PAGE to initiate data transfers to Packet RAM. This bit is intended for use with DBG_PAGE=13, but is available on all pages. Its usage is optional.
DBG_RAM_FULL[1:0]	r	Packet RAM Debug RAM Full. Status Bits indicating that Packet RAM0 (or RAM1) is full, and the Packet RAM Debug engine has

Table continues on the next page...

Field	R/W	Description
		attempted to write another word to that RAM. This, and any subsequent write attempts, will not alter the contents of the RAM, once it has reached capacity. To clear the DBG_RAM_FULL[1:0] bits, the DBG_PAGE must be set to 0 and then set to a non-zero value (which starts a new capture by the Packet RAM Debug engine). That is, after a capture completes, the DBG_RAM_FULL[1:0] bits only reset to zero upon the start of the subsequent capture, at which time it is safe for software to start polling these bits again. **NOTE:** If a Debug Stop Trigger is selected (DBG_STOP_TRG > 0), writes to RAM do not stop when DBG_RAM_FULL bit(s) become set; instead, writes continue after an address wrap-around to 0, and will continue indefinitely until either a stop trigger condition occurs, or the debug session is ended by taking DBG_PAGE=0. The DBG_RAM_FULL[1:0] remain a valid indicator that the RAM(s) has reached full status at least once during the debug session.
DBG_SOFT_INFO_SEL	rw	Packet RAM Debug PHY Soft Info Output Selector. When acquiring data in Packet RAM Debug Mode with the DBG_PAGE=DMDSOFT page, this bit selects the signal from the PHY used to capture the soft decision data. **NOTE:** When the soft bits are captured without regard to aa_sfd_matched, the bits will be sampled at a higher rate before Access Address detection. Prior to the assertion of aa_sfd_matched, the soft bits are at OSR 8. Following the assertion of aa_sfd_matched plus 2, the soft bits are at OSR 2. 0: PHY output cg_vbr_en is used to capture soft decision data 1: PHY output fsk_demod_bit_valid is used to capture soft decision data
DBG_START_TRG[2:0]	rw	The Packet RAM Debug Start Trigger, if desired, can be selected from the sources in the following table: 0: no trigger 1: FSK PHY: gfsk_preamble_found 2: FSK PHY: aa_sfd_matched 3: Reserved 4: Reserved 5: RXDIG: agc_dcoc_gain_chg 6: TSM: rx_dig_en 7: TSM: tsm_spare2_en
DBG_START_EDGE	rw	Packet RAM Debug Start Trigger Edge Selector. DBG_START_EDGE selects the edge sensitivity (rising or falling) of the selected start-trigger. 0: Trigger fires on a rising edge of the selected trigger source 1: Trigger fires on a falling edge of the selected trigger source
DBG_STOP_TRG[3:0]	rw	Packet RAM Debug Stop Trigger Selector. The Packet RAM Debug Stop Trigger, if desired, can be selected from the sources in the following table: 0: no trigger 1: FSK PHY: gfsk_preamble_found 2: FSK PHY: aa_sfd_matched 3: Reserved

Table continues on the next page...

Field	R/W	Description
		4: Reserved 5: RXDIG: agc_dcoc_gain_chg 6: TSM: rx_dig_en 7: TSM: tsm_spare3_en 8: TSM: pll unlock 9: BLE: crc_err_count_incr (increment DTM RX bad packet counter) 10: CRC FAIL: GEN_FSK only 11: HEADER FAIL: GEN_FSK only 12-15: Reserved
DBG_STOP_EDGE	rw	DBG_STOP_EDGE selects the edge sensitivity (rising or falling) of the selected stop-trigger. 0: Trigger fires on a rising edge of the selected trigger source 1: Trigger fires on a falling edge of the selected trigger source
DBG_START_TRIGGERED	r	Packet RAM Debug Start Triggered. This read-only status bit becomes set, when, during a Packet RAM Debug session (DBG_PAGE > 0), the trigger source selected by DBG_START_TRG[2:0] occurs, with the edge sensitivity selected by DBG_START_EDGE. To clear this bit, set DBG_PAGE=0.
DBG_STOP_TRIGGERED	r	Packet RAM Debug Stop Triggered. This read-only status bit becomes set, when, during a Packet RAM Debug session (DBG_PAGE > 0), the trigger source selected by DBG_STOP_TRG[3:0] occurs, with the edge sensitivity selected by DBG_STOP_EDGE. To clear this bit, set DBG_PAGE=0.

45.3.8.3 Functional Description

45.3.8.3.1 Transceiver DMA Mode

Tranceiver DMA mode takes internal receive data from the RX_DIG or PHY, packs the data into an optimal format for bandwidth or RAM dimensions, and hands off the data to an external (SoC) DMA controller for subsequent transfer to system memory. Control and configuration of transceiver DMA operation is governed by the bit fields in the DMA_CTRL register of XCVR address space. These bit fields are referenced in the description that follows. In a typical transceiver DMA debug session, the transceiver DMA hardware performs the following steps:

1. Selects a source of internal data from RX_DIG or PHY for capture based on DMA_PAGE[3:0] setting
2. Waits for a hardware start-trigger, if so configured
3. Uses a defined capture signal from RX_DIG or PHY to latch the internal data into a holding register
4. Sequentially packs 2 or more captures of internal data into a 32-bit register until the packing requirements for that DMA_PAGE setting are met

5. Uses the radio's SRS DarkBlue Line-compliant DMA interface to notify SoC DMA controller there is data ready for transfer
6. Repeats steps 3-5 until the debug session is complete (DMA_PAGE set to 0).

Each source of RX_DIG or PHY data has a capture signal associated with it. The capture signal is generated in the source block. The capture signal is a one-reference-clock wide pulse that indicates that the next sample (or samples) of source data is ready for capture. For RX_DIG data sources, the frequency of the capture signal is affected by the RX_DECT_FILT_OSR[2:0] (over-sampling rate) setting. Due to system bandwidth constraints not all RX_DECT_FILT_OSR settings are available, and in some cases the ideal mission mode setting can't be used. The maximum RX_DECT_FILT_OSR setting is dependent on DMA_PAGE, and is listed in the table below. For any DMA_PAGE selecting source data from RX_DIG, set RX_DIG_CTRL[RX_DMA_DTEST_EN]=1 (in mission mode, this bit is clear to reduce power consumption.)

At the assertion of the capture signal, the transceiver takes the data sample(s) presented by RX_DIG or PHY and stores them in a holding register. Subsequent assertions on the capture signal fill different portions of the holding register, until the packing requirements for the selected DMA_PAGE setting are acheived. A description of the packing requirements for each DMA_PAGE setting is included in the table below. Packing is complete when the page-specific packing requirements are met, and the contents of the holding register are transferred into the DMA_DATA[31:0] register for pickup by the SoC DMA controller.

The DMA_DATA[31:0] is a memory-mapped, read-only IPS bus register. It is available for host access, but its primary function is to be read only by the DMA controller upon request from the radio. Once valid data is stored into the DMA_DATA register, the radio notifies the SoC DMA controller that a new sample is ready using one of 2 methods:

REQUEST-PER-SAMPLE Mode

In this mode, the radio signals the external DMA controller with a new transfer request whenever a new 32-bit sample word is ready in DMA_DATA[31:0]. This is accomplished by setting SINGLE_REQ_MODE=0. The maximum DMA transfer rate in this mode is 2Msamples/sec. This mode is not optimal from the point-of-view of the DMA controller, because first-cycle-latency is incurred for every sample. The debug session block size (total data to be tranferred) need not be known in advance. To reduce IPS bus latency in this mode, in order to meet the 2Msamples/sec data rate, the following programming restrictions are mandatory:

1. Both radio and MCU must be programmed to run off the same clock source (no clock domain crossing)
2. That clock source must be the Reference Oscillator.

3. Synchronization of the DMA request and done signals must be bypassed, so set BYPASS_DMA_SYNC=1.
4. Synchronization of the IPS bus must be bypassed, so set RSIM_CTRL[GASKET_BYPASS_OVRD_EN]=RSIM_CTRL[GASKET_BYPASS_OVRD]=1.
5. MCU code should be executed out of internal RAM, not flash, since flash access requirements might not be met at this bus frequency.
6. No Radio IPS bus activity by the host is allowed during a DMA session
7. Once DMA debug session is complete, set RSIM_CTRL[GASKET_BYPASS_OVRD_EN]=RSIM_CTRL[GASKET_BYPASS_OVRD]=0.
8. Radio IPS accesses are now permitted

Note: From the SoC perspective, steps (1) and (2) put the SoC in a configuration that is a special case where full MCU-to-radio IPS bus timing has not been closed. In this mode, only paths associated with, and required for, DMA transfers, have closed timing. Therefore, no other IPS bus or ordinary mission-mode activity should be attempted in this mode. This restriction means host accesses to all radio registers is prohibited.

In REQUEST-PER-SAMPLE mode, whenever a new sample is ready, the radio asserts **ipd_req_radio_rx** to notify the external DMA controller. The controller will respond to the request, by reading the sample from the DMA_DATA register, and then asserts **ipd_done_radio_rx**. In order to ensure that the 2Msample/second transfer rate is met, the controller must respond with **ipd_done_radio_rx** within 15 reference oscillator clock periods after the radio asserts **ipd_req_radio_rx**. This timing is acheivable if the aforementioned configuration instructions are followed. IC-level simulation of this mode demonstrates that request-to-done timing is a maximum of 13 reference clock periods, after proper configuration. As long as the external controller asserts **ipd_done_radio_rx** no later than 14 reference clock periods after the radio asserts **ipd_req_radio_rx**, the radio shall deassert **ipd_req_radio_rx** on the next reference clock cycle, and hold it deasserted until a new sample is ready. If the external controller asserts **ipd_done_radio_rx** exactly 15 clock periods after the radio asserts **ipd_req_radio_rx**, the radio shall deassert **ipd_req_radio_rx** after the 14th consecutive clock cycle of **ipd_req_radio_rx** assertion, for one clock cycle, in order to generate a low-to-high transition on **ipd_req_radio_rx** on the following clock cycle, when a new sample will be ready for transfer. This is the zero-operating-margin case. If the external controller does not assert **ipd_done_radio_rx** within 15 reference clock periods, there is no guarantee that the controller retrieved the sample from DMA_DATA in time, before the radio updated DMA_DATA to the next sample. This is considered a system malfunction, yielding the possibility that samples have been missed by the controller. It is not a certainty that samples were skipped; it depends on the timing of the DMA IPS bus read relative to the request, but this condition should be avoided.

SINGLE REQUEST Mode

In Single Request mode, the entire block of desired DMA data, is transferred to system memory based on a single request by the transceiver to the external DMA controller. The block size of the debug session must be known in advance, and this size must be programmed into the DMA controller. In this mode, DMA controller does not incur first-cycle latency on every sample, only on the first sample of a debug session. This is because there is only a single, initial request from the radio. In this mode, the MCU can be configured for any clock source, DMA-signal synchronization is not required (BYPASS_DMA_SYNC=0), and RSIM IPS gasket bypass is also not required. Host IPS accesses to radio registers propagate through the synchronizing IPS bus gaskets as in mission mode; however IPS bus accesses to transceiver registers during DMA activity should be avoided in order to avoid delaying the controller accesses to the DMA_DATA register.

In this mode, once the radio has its first sample packed into DMA_DATA and ready to transfer to the external controller, it will assert **ipd_req_radio_rx**. The radio will hold **ipd_req_radio_rx** asserted until the first read access to the DMA_DATA register by the external controller. At this point, the radio shall deassert **ipd_req_radio_rx** and hold it low for the remainder of the debug session (until DMA_PAGE is set to 0). The external controller shall transfer the entire block of DMA data based on this single radio request. The radio shall ignore **ipd_done_radio_rx** in this mode. The external controller, operating with reduced latency in this mode, lacking the need to process new requests for every sample, will be able to consume samples from the radio faster than the radio can generate them (i.e., greater than 2Msamples/sec). Thus, the radio will assert **ips_xfr_wait_xcvr** to the controller to pace the transfers. When the external controller reads a sample from DMA_DATA, the radio will assert **ips_xfr_wait_xcvr** on the next reference clock cycle, and keep it asserted until a new sample is available on DMA_DATA; it will then deassert **ips_xfr_wait_xcvr** for 1 clock cycle to allow the controller to access DMA_DATA, then reassert it again. This process iterates until the debug session is complete: the pre-programmed block of data is transferred, and DMA_PAGE is set to 0 by program software.

The IPS SkyBlue signal **ips_xfr_wait** allows a peripheral to insert wait states on the IPS bus for cases in which it cannot deliver readback data (ips_rdata) in the same clock cycle that ips_module_en is asserted. The transceiver DMA interface uses **ips_xfr_wait_xcvr** to throttle the DMA transfers as described above. Because of the potential hazards that can arise due to an extended assertion of **ips_xfr_wait** on any IPS bus, which could result from an incorrectly programmed transceiver or DMA controller, a programmable timeout mechanism has been built in to the transceiver DMA hardware. The timeout mechanism measures the amount of time **ips_xfr_wait_xcvr** is asserted, in microseconds, by the transceiver DMA hardware, and compares it to the contents of the DMA_TIMEOUT[3:0] register, which is also calibrated in microseconds. When they match, the transceiver

DMA will force **ips_xfr_wait_xcvr** low, and set the DMA_TIMED_OUT status bit. To clear this bit, disable transceiver DMA by writing DMA_PAGE=0, then write 1 to DMA_TIMED_OUT bit to clear it. Since the timeout condition arose from an erroneously configured transceiver or DMA controller, the data transferred during the DMA session, if any, should be considered invalid. Troubleshoot and rectify the conditions which led to the timeout condition, before initiating a new DMA debug session. DMA timeout applies only to Single Request mode, since **ips_xfr_wait_xcvr** is only asserted in this mode.

PAGE TABLE

The following table describes the available DMA_PAGE settings. The internal source data selected, the capture signal employed, the data packing method, the OSR setting, and the data rate, are provided for each DMA_PAGE setting.

DMA PAGE	MNEMONIC	PACKING	OSR	CAPTURE SIGNAL	REQUEST RATE	REMARKS
0	DMAIDLE	---	---	---	---	Transceiver DMA Idle. No DMA Activity.
1	RXDIGIQ	{4'd0, rx_dig_q[11:0], 4'd0, rx_dig_i[11:0]}	4	rx_dig_iq_vld (2MHz)	2MHz	Q and I data captured. Max supported sample rate ref_osc/16 =2MHz for 32MHz clock.
2	RXDIGI	{4'd0, rx_dig_i[11:0], 4'd0, rx_dig_i[11:0]}	2	rx_dig_iq_vld (4MHz)	2MHz	I data data captured. Max supported sample rate ref_osc/8 = 4MHz for 32MHz clock. Newer sample is in upper bytes.
3	RXDIGQ	{4'd0, rx_dig_q[11:0], 4'd0, rx_dig_q[11:0]}	2	rx_dig_iq_vld (4MHz)	2MHz	Q data captured. Max supported sample rate ref_osc/8 = 4MHz for 32MHz clock. Newer sample is in upper bytes.
4	RAWADCIQ	{rx_dig_q[7:0], rx_dig_i[7:0], rx_dig_q[7:0], rx_dig_i[7:0]}	2	rx_dig_iq_vld (4MHz)	2MHz	Raw ADC capture of I and Q channel but every other I and Q ADC sample are discarded The

Table continues on the next page...

DMA PAGE	MNEMONIC	PACKING	OSR	CAPTURE SIGNAL	REQUEST RATE	REMARKS
						vld strobe will assert at ref_clock/8 = 4MHz rate. (OSR needs to be programmed for 16). Newest samples are in upper bytes
5	RAWADCI	{rx_dig_i[7:0], rx_dig_i[7:0], rx_dig_i[7:0], rx_dig_i[7:0]}	1	rx_dig_iq_vld (8MHz)	2MHz	Raw ADC capture of I channel. The vld strobe will assert at ref_clock/4 = 8MHz rate. (OSR needs to be programmed for 8). Newest sample is in MSB
6	RAWADCQ	{rx_dig_q[7:0], rx_dig_q[7:0], rx_dig_q[7:0], rx_dig_q[7:0]}	1	rx_dig_iq_vld (8MHz)	2MHz	Raw ADC capture of Q channel. The vld strobe will assert at ref_clock/4 = 8MHz rate. (OSR needs to be programmed for 8). Newest sample is in MSB
7	DCESTIQ	{4'd0, Qdc_est[11:0], 4'd0, Idc_est[11:0]}	4	IQdc_est_vld (2MHz)	2MHz	Q and I DCOC tracking estimator data captured. Max supported sample rate ref_osc/16 =2MHz for 32MHz clock
8	DCESTI	{4'd0, Idc_est[11:0], 4'd0, Idc_est[11:0]}	2	IQdc_est_vld (4MHz)	2MHz	I channel DCOC tracking estimator data captured. Max supported sample rate ref_osc/8 = 4MHz for 32MHz clock. Newer sample is in upper bytes
9	DCESTQ	{4'd0, Qdc_est[11:0],	2	IQdc_est_vld (4MHz)	2MHz	Q channel DCOC tracking estimator data

Table continues on the next page...

DMA PAGE	MNEMONIC	PACKING	OSR	CAPTURE SIGNAL	REQUEST RATE	REMARKS
		4'd0, Qdc_est[11:0]}				captured. Max supported sample rate ref_osc/8 = 4MHz for 32MHz clock. Newer sample is in upper bytes
10	RXINPH	{3'd0,rxin_ph[4:0], 3'd0, rxin_ph[4:0], 3'd0, rxin_ph[4:0], 3'd0, rxin_ph[4:0]}	1	rx_dig_ph_vld (8MHz)	2MHz	Phase data captured. Max sample rate is ref_osc/2 = 8MHz for 32MHz clock. Newest sample in MSB
11	DMDHARD	32 x {fsk_demod_bit}	---	cg_vbr_en (1MHz)	31.250KHz	Demodulated bit stream, hard decision output, captured at 1Mb/sec and packed into a 32-bit word with the newest sample in the MS bit position
12	DMDSOFT	{7'd0, fsk_demod_bit,fsk_demod_soft[7:0], 7'd0, fsk_demod_bit,fsk_demod_soft[7:0]}	---	cg_vbr_en (1MHz)	0.5MHz	Soft decision output, and hard decision output, captured at 1Mb/sec. Newest sample in the MS byte (bit)
13	DMDDATA	32 x {data_out}	---	cg_vbr_en (1MHz)	31.250KHz	Demodulated bit stream, post-FIFO, captured at 1Mb/sec and packed into a 32-bit register with the newest sample in the MS bit position.
14	CFOPHASE	{3'd0, cfo_corrected_ph_in[4:0], 3'd0, cfo_corrected_ph_in[4:0], 3'd0, cfo_corrected_ph_in[4:0], 3'd0, cfo_corrected_ph_in[4:0]}	---	rx_dig_ph_vld (8MHz)	2MHz	CFO-corrected phase, captured at rx_dig_ph_vld (8MHz). Newest sample in MS byte

EXAMPLE PROCEDURE

The following steps outline the setup and execution of a Transceiver DMA debug session. This example uses BLE protocol, acquires RX DIG I/Q data, and employs REQUEST-PER-SAMPLE DMA mode:

1. Configure MCU to run off the RF OSC clock source, with divider ratio = 1 (no division)
2. Program RX_DIG for BLE protocol, including RX_DIG_CTRL[RX_DECT_FILT_OSR]=4
3. Set RX_DIG_CTRL[RX_DMA_DTEST_EN]=1
4. Enable clocking to the DMA engine by setting DMA_CTRL[DMA_EN]=1
5. Select a DMA start-trigger source, if desired, by setting DMA_CTRL[DMA_START_TRG] to a non-zero value
6. Program DMA_CTRL[DMA_PAGE]=RXDIGIQ, DMA_CTRL[SINGLE_REQ_MODE]=0, and DMA_CTRL[BYPASS_DMA_SYNC]=1.
7. Set TSM_CTRL[FORCE_RX_EN]=1 to launch an RX sequence.
8. Set RSIM_CTRL[GASKET_BYPASS_OVRD_EN]=RSIM_CTRL[GASKET_BYPASS_OVRD]=1 (XCVR IPS gasket bypass)
9. Begin transmitting RF to the DUT
10. Wait for sufficient RX data to be collected to satisfy the requirements of the debug session
11. Clear RSIM_CTRL[GASKET_BYPASS_OVRD_EN]=RSIM_CTRL[GASKET_BYPASS_OVRD]=0 (XCVR IPS gasket bypass off)
12. Program DMA_CTRL[DMA_PAGE]=0 to terminate the acquisition
13. Download the captured data from system memory

Note: Steps 5 (establish the start trigger) and 6 (select the DMA_PAGE to enable the DMA engine), should be performed as sequential, discrete writes to DMA_CTRL, and not combined into a single write.

45.3.8.3.2 Packet RAM Debug Mode

Packet RAM Debug mode takes internal receive data from the RX_DIG or PHY, packs the data into a format suitable for storage in the radio's Packet RAM, and then directly fills the 2 RAMs with source data, under program control. Control and configuration of Packet RAM Debug operation is governed by the bit fields in the PACKET_RAM_CTRL

register of XCVR address space. These bit fields are referenced in the description that follows. In a typical Packet RAM debug session, the debug hardware performs the following steps:

1. Selects a source of internal data from RX_DIG or PHY for capture based on DBG_PAGE[3:0] setting
2. Waits for a hardware start-trigger, if so configured
3. Uses a defined capture signal from RX_DIG or PHY to latch the internal data into a holding register
4. Selects a RAM fill method: sequential or simultaneous, based on DBG_PAGE setting
5. Sequentially packs 2 or more captures of internal data into a 16- or 32-bit register until the packing requirements for that DBG_PAGE setting are met
6. Directly accesses the 2 Packet RAM instances to store the receive data (addressing, write enabling, chip enabling, etc.)
7. Repeats steps 3-6 until the debug session is complete (DBG_PAGE set to 0), or a stop-on-trigger event occurs

Each source of RX_DIG or PHY data has a capture signal associated with it. The capture signal is generated in the source block. The capture signal is a one-reference-clock wide pulse that indicates that the next sample (or samples) of source data is ready for capture. For RX_DIG data sources, the frequency of the capture signal is affected by the RX_DECT_FILT_OSR[2:0] setting. In general, there are no restrictions on the OSR setting for Packet RAM debug mode; the RAM fill rate should be able to track with the mission mode OSR settings used by the supported protocols. For any DMA_PAGE selecting source data from RX_DIG, set RX_DIG_CTRL[RX_DMA_DTEST_EN]=1 (in mission mode, this bit is clear to reduce power consumption.)

At the assertion of the capture signal, the transceiver takes the data sample(s) presented by RX_DIG or PHY and stores them in a holding register. Subsequent assertions on the capture signal fill different portions of the holding register, until the packing requirements for the selected DBG_PAGE setting are acheived. Packing of the receive data takes into account the RAM fill method employed, based on the DBG_PAGE setting. RAM fill can be sequential (RAM0 fills first completely, followed by RAM1) or simultaneous (both RAMs fill concurrently). A description of the packing requirements and the RAM fill method for each DBG_PAGE setting is included in the table below. Packing is complete when the page-specific packing requirements are met, and the contents of the holding register are written directly to the RAM instance(s).

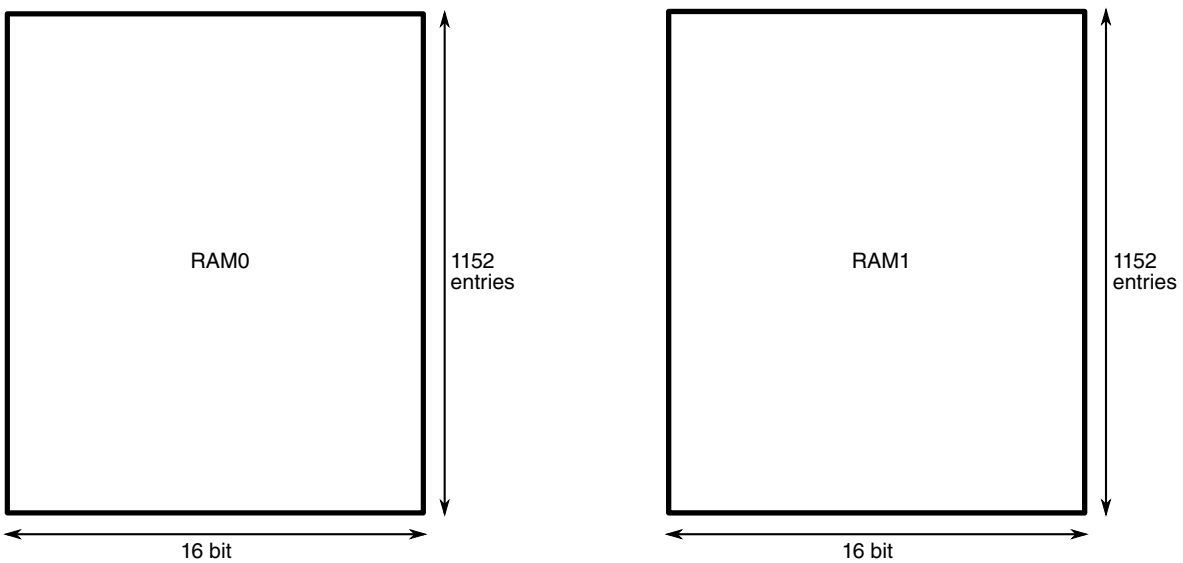
The bandwidth to Packet RAM is much higher than Transceiver DMA, since the overhead of DMA-related synchronization and multi-master arbitration can be avoided. The maximum transfer rate for RAM debug mode, for a sample of width less than 8 bits (e.g., RAW ADC samples), and simultaneous RAM fill, can be as high as

128Msamples/sec (compared to 2Msamples/sec for DMA). The tradeoff is that the total Packet RAM space is smaller than the space available in system memory, restricting the size of the block of data that can be saved.

PACKET RAM

The Packet RAM consists of 2 identical RAM blocks, with identical dimensions, as shown in the following diagram.

RAM DIMENSIONS
2304B (RAM0) + 2304B (RAM1) = 4608B (total)



The RAMs are referred to as RAM0 and RAM1. In mission mode, the RAMs are partitioned into segments to suit the needs of the various protocols, and the sectioning is protocol-dependent. However for the purpose of Packet RAM Debug mode, the entire RAM space is made available for receiver data storage, as one contiguous address space. In Packet RAM Debug mode, the debug engine commandeers both RAMs for buffering receive data, and access attempts by the (mission mode) protocol engines to the RAMs are blocked. For a Packet RAM Debug session, set XCVR_RAM_ALLOW=1. This affords the transceiver IPS bus and the RAM debug engine exclusive access to both RAM blocks comprising the Packet RAM. It also continuously asserts the Chip Enable to both RAMs, and enables a free-running clock to both RAMs, overriding these controls from the protocol link layers.

A Packet RAM debug session consists of 2 stages: Acquisition and Downloading.

ACQUISITION

In a typical debug session, the selection of a receive data source is made by setting DBG_PAGE to a non-zero value (see the table below for a description of the debug pages). Setting DBG_PAGE to a non-zero value activates the Packet RAM Debug Engine. The engine has write-only access to both RAM's while receive data is being collected. (XCVR host IPS access is not blocked, but should be avoided during acquisition).

If a hardware start-trigger is selected (DBG_START_TRG > 0), the debug engine will wait until the selected triggering event occurs before acquiring any samples; otherwise, acquisition begins as soon as DBG_PAGE is set to a non-zero value.

The RAM's will fill with receive data, either sequentially or simultaneously, determined by the DBG_PAGE setting. If the receive data source selection involves simultaneous capture of symmetric I and Q channel data, RAM0 is dedicated to I channel capture and RAM1 to Q channel; in such cases, both RAMs fill simultaneously. In all other cases, the RAMs fill sequentially: RAM0 fills to capacity, followed by RAM1. RAM addressing always starts at address 0 of either (or both) RAMs; the address for both RAMs is always reset to 0 when DBG_PAGE=0.

If a stop-trigger is not selected (DBG_STOP_TRG = 0): When a RAM reaches capacity, further write attempts by the debug engine to that RAM are blocked, and the DBG_RAM_FULL[0] or [1] status bits become set, depending on which RAM is at capacity. When the fill method is simultaneous, both RAMs will reach the full state at the same instant. Otherwise, RAM0 will reach full first. When sufficient receive data has been acquired to satisfy the requirements of the debug session, or when both RAMs are full, the acquisition should be terminated by setting DBG_PAGE=0. The DBG_RAM_FULL[0] and [1] status bits are "sticky"; once set, they will remain set even after DBG_PAGE is set to 0 to end the session. Both bits will self-clear when a new acquisition is started by setting DBG_PAGE to a non-zero value. The XCVR_RAM_ALLOW bit should be set for the duration of the debug session.

If a stop-trigger is selected (DBG_STOP_TRG > 0): When a RAM (or both RAMs) reaches capacity, further write attempts by the debug engine to that RAM are not blocked, but the DBG_RAM_FULL[0] or [1] status bits do become set. After capacity is reached on both, or either RAM, depending on the fill method, the debug engine will wrap back to address 0 of the RAM (or RAM pair) and continue filling, until the selected stop-trigger event occurs. When the stop-trigger event occurs, the debug engine will stop filling the RAMs immediately, the DBG_STOP_TRIGGERED status bit will become set, and the registers RAM0_STOP_ADDRESS[10:0] and RAM1_STOP_ADDRESS[10:0] will hold the addresses of the last RAM word, for both RAMs, filled with a valid sample before the trigger. At this point, the acquisition should be terminated by setting DBG_PAGE=0. The DBG_STOP_TRIGGERED and RAMx_STOP_ADDRESS registers are "sticky"; they will hold their state even after DBG_PAGE is set to 0 to end

the session. DBG_STOP_TRIGGERED will self-clear, and RAMx_STOP_ADDRESS will reset to 0, when a new acquisition is started by setting DBG_PAGE to a non-zero value. The XCVR_RAM_ALLOW bit should be set for the duration of the debug session.

A diagram of the acquisition process is shown below:

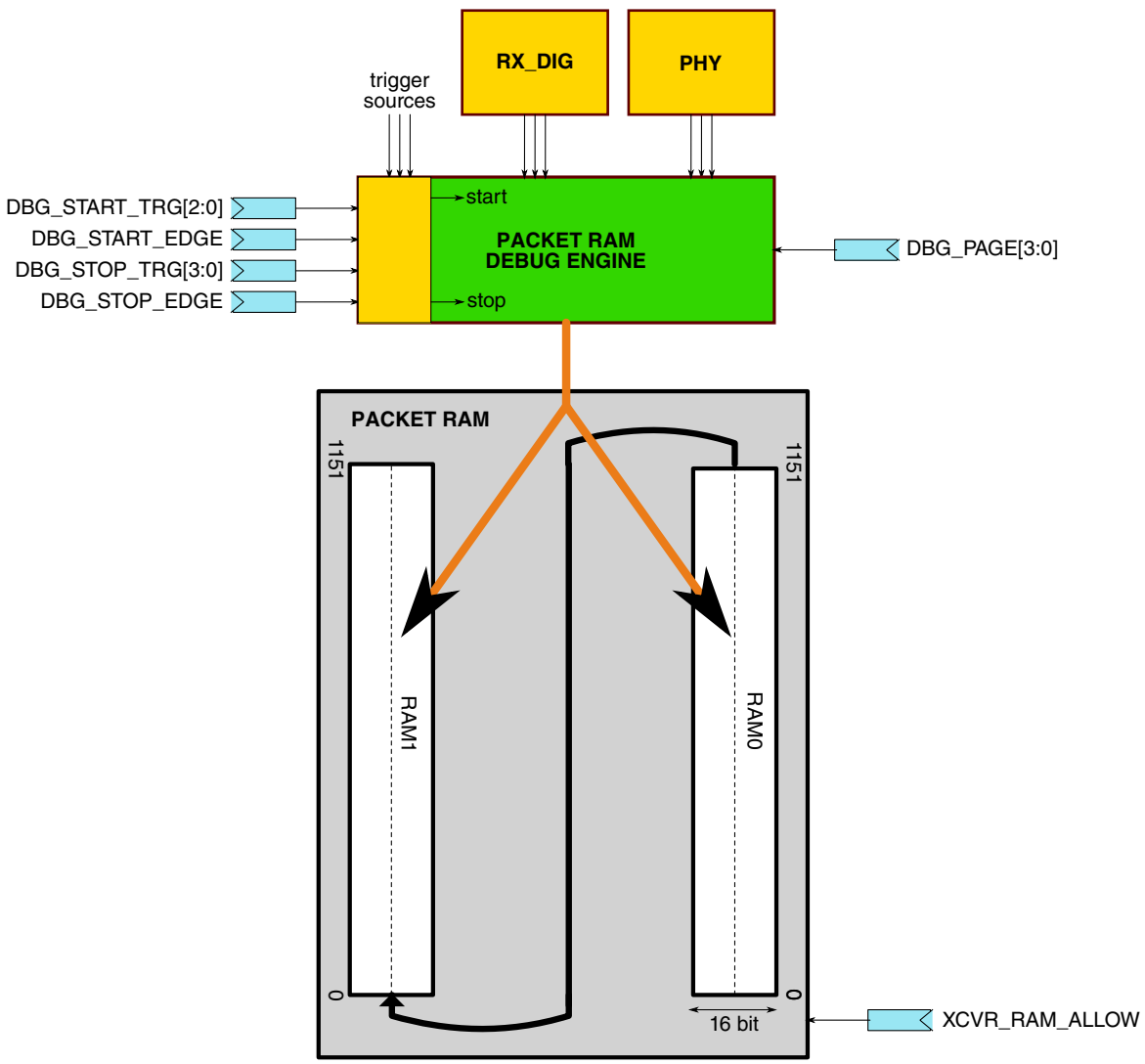


Figure 45-97. Packet RAM Debug Mode Block Diagram

DOWNLOADING

As soon as sufficient data has been collected to satisy the requirements of the debug session, DBG_PAGE should be set to 0 to end the acquisition. The receive data stored in RAM may now be downloaded from the Packet RAM for analysis and post-processing.

XCVR_RAM_ALLOW should stay set to allow XCVR IPS access to the RAM. The receive data in the RAM has been packed and distributed between the 2 RAMs in accordance with the DBG_PAGE selection.

Regardless of the DBG_PAGE selection, the host accesses the RAM between addresses XCVR_BASE+0x700 (base of RAM0 or RAM1) and XCVR_BASE+0xFFF (last address of RAM0 or RAM1). Which RAM block gets mapped into this space depends on the setting of the PACKET_RAM_CTRL[XCVR_RAM_PAGE] bit. Set XCVR_RAM_PAGE=1 to access RAM1 in this space; otherwise RAM0 is accessed in this space.

A diagram depicting how the 2 RAM instances are situated within XCVR address space is shown below.

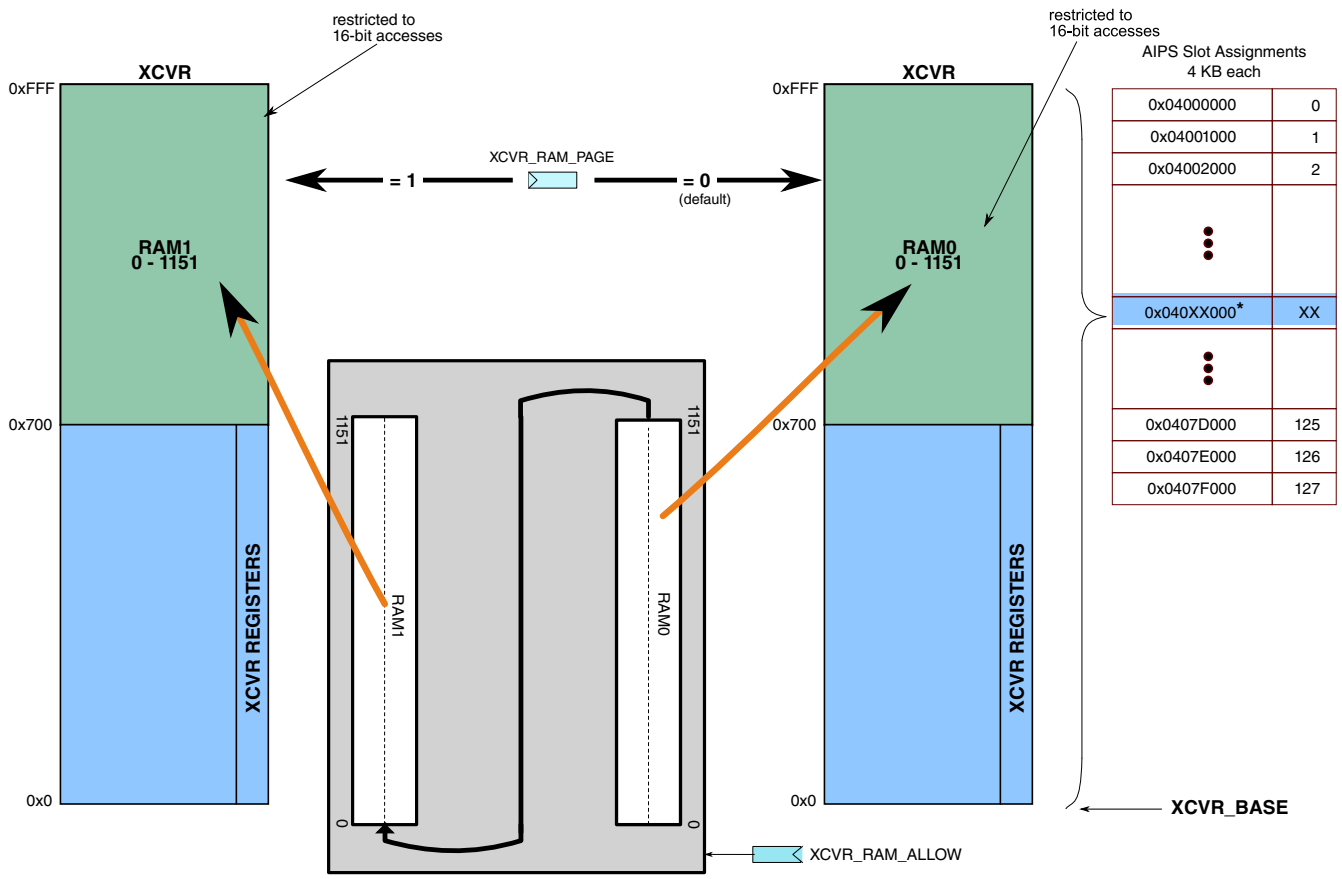


Figure 45-98. Packet RAM Memory Mapping in XCVR Space

PAGE TABLE

The following table describes the available DBG_PAGE settings. In most cases, the DBG_PAGE settings track the Transceiver DMA's DMA_PAGE settings; however DMA_PAGE settings which capture I- and Q-channel-only data are redundant for RAM

debug mode, because the bandwidth to packet RAM is high enough to capture I and Q simultaneously with no loss of samples. The internal source data selected, the capture signal employed, the data packing method, and the RAM fill method, and the data rate, are provided for each DBG_PAGE setting.

DBG PAGE	MNEMONIC	PACKING	CAPTURE SIGNAL	FILL METHOD	REMARKS
0	DBGIDLE	---	---	---	RAM Debug DMA Idle. No RAM Debug Activity.
1	RXDIGIQ	{4b0, rx_dig_i[11:0]} → RAM0 {4b0, rx_dig_q[11:0]} → RAM1	rx_dig_iq_vld	Simul	Q and I data captured simultaneously.
2	---	---	---	---	---
3	---	---	---	---	---
4	RAWADCIQ	{rx_dig_i[7:0],rx_dig_i[7:0]} → RAM0 {rx_dig_q[7:0],rx_dig_q[7:0]} → RAM1	rx_dig_iq_vld	Simul	Raw ADC capture of I and Q channel. Newest samples are in upper bytes
5	---	---	---	---	---
6	---	---	---	---	---
7	DCESTIQ	{4b0, Idc_est[11:0]} → RAM0 {4b0, Qdc_est[11:0]} → RAM1	IQdc_est_vld	Simul	Q and I DCOC tracking estimator data captured.
8	---	---	---	---	---
9	---	---	---	---	---
10	RXINPH	{3'd0,rxin_ph[4:0], 3'd0,rxin_ph[3:0]}	rx_dig_ph_vld	Sequential	Phase data captured. Newest sample in MSB. Fill RAM0 first with 16 bit words, then fill RAM1
11	DMDHARD	16 x {fsk_demod_bit}	cg_vbr_en (1MHz)	Sequential	Demodulated bit stream, hard decision output, captured at 1Mb/sec and packed into a 16-bit word with the newest sample in the MS bit position
12	DMDSOFT	{6'd0, aa_sfd_matched, fsk_demod_bit,	cg_vbr_en (1MHz)	Sequential	Soft decision output, and hard decision output, captured at 1Mb/

Table continues on the next page...

DBG PAGE	MNEMONIC	PACKING	CAPTURE SIGNAL	FILL METHOD	REMARKS
		fsk_demod_soft[7:0]}			sec. Newest sample in the MS byte (bit)
13	DMDDATA	16 x {data_out}	cg_vbr_en (1MHz)	Sequential	Demodulated bit stream, post-FIFO, captured at 1Mb/sec and packed into a 16-bit register with the newest sample in the MS bit position. Capturing begins only after AA match *and* the first PHY data_out_valid is asserted.
14	CFOPHASE	{3'd0, cfo_corrected_ph_in[4:0], 3'd0, cfo_corrected_ph_in[4:0]}	rx_dig_ph_vld	Sequential	CFO-corrected phase, captured at rx_dig_ph_vld. Newest sample in MS byte

EXAMPLE PROCEDURE

The following steps outline the setup and execution of a Packet RAM debug session. This example uses BLE protocol, and acquires RX DIG I/Q data

1. Program RX_DIG for BLE protocol
2. Set RX_DIG_CTRL[RX_DMA_DTEST_EN]=1
3. Set PACKET_RAM_CTRL[XCVR_RAM_ALLOW]=1 to allow exclusive XCVR access to radio RAMs
4. Enable clocking to the Packet RAM Debug engine by setting PACKET_RAM_CTRL[DBG_EN]=1
5. Select a DBG start-trigger source, if desired, by setting PACKET_RAM_CTRL[DBG_START_TRG] to a non-zero value
6. Select a DBG stop-trigger source, if desired, by setting PACKET_RAM_CTRL[DBG_STOP_TRG] to a non-zero value
7. Set TSM_CTRL[FORCE_RX_EN]=1 to launch an RX sequence.
8. Begin transmitting RF to the DUT
9. Program PACKET_RAM_CTRL[DBG_PAGE]=RXDIGIQ to begin acquisition
10. **If a stop-trigger is selected (DBG_STOP_TRG > 0) --**
11. Wait for DBG_STOP_TRIGGERED=1, indicating a stop-trigger occurred
12. **If a stop-trigger is not selected (DBG_STOP_TRG = 0) --**
13. Wait for sufficient RX data to be collected, or poll DBG_RAM_FULL[0] (and/or DBG_RAM_FULL[1]), to satisfy the requirements of the debug session

14. Program PACKET_RAM_CTRL[DBG_PAGE]=0 to terminate the acquisition
15. Download the captured data from Packet RAM

Note: Steps 3, 4, 5, and 6 can be combined into a single write to PACKET_RAM_CTRL.

Note: Step 9 (select the DBG_PAGE to enable the DBG engine) should be performed as a sequential, discrete write, after steps 5 and/or 6 (establishing the start/stop triggers) and not combined into a single write to PACKET_RAM_CTRL.

45.3.8.3.3 Simultaneous Transceiver DMA and RAM Debug

Transceiver DMA mode and Packet RAM Debug mode are described in the previous sections. The DMA and Debug engines are separate, independent entities, which theoretically allows both to be employed simultaneously. The most practical case for simultaneous operation would be to select an RX_DIG source for DMA_PAGE and a PHY source for DBG_PAGE. During a such simul debug session, RX_DIG data would go to system memory and PHY data to Packet RAM. After acquisition, data would be downloaded from both memories. Post-processing software could consolidate the data, to take advantage of the synergy that results from combining data captured simultaneously from multiple sources. (DTEST allows a possible third simultaneous source, beyond the scope of this document).

In addition, it is also theoretically possible to combine either or both debug modes with mission-mode packet reception, with the caveat that the Packet RAM would not be available to receive mission-mode packet data in such a scenario. (Packet processing, filtering, CRC checking, and Link Layer interrupts don't rely on packet RAM and so would not be impacted by the debug modes). Such a combination has been demonstrated successfully in radio-level simulation, with the following setup:

1. The GENERIC_FSK Link Layer is engaged in RX mode
2. Transceiver DMA mode is engaged with DMA_PAGE set to RXINPH (RX phase capture)
3. Packet RAM Debug mode is engaged with DBG_PAGE set to DMDSOFT (PHY soft decision data)
4. A GENERIC_FSK packet is received, CRC passes, and GENERIC_FSK RX_IRQ and SEQ_END_IRQ are asserted (GENERIC_FSK packet data not stored)
5. RX phase data is downloaded from system memory and verified
6. Demod soft decision data is downloaded from Packet RAM and verified

45.3.8.3.4 Trigger Sources

START TRIGGER SOURCES

Registers DMA_CTRL[DMA_START_TRG] and PACKET_RAM_CTRL[DBG_START_TRG], enable a hardware trigger that must assert before acquisition begins, for the transceiver DMA and Packet RAM debug engines, respectively. The same description applies to both registers (i.e., trigger sources are the same), as indicated in the following table:

xxx_START_TRG[2:0]	HARDWARE TRIGGER
0	no trigger
1	FSK PHY: gfsk_preamble_found
2	FSK PHY: aa_sfd_matched
3	Reserved
4	Reserved
5	RXDIG: agc_dcoc_gain_chg
6	TSM: rx_dig_en
7	TSM: tsm_spare2_en

STOP TRIGGER SOURCES

Register PACKET_RAM_CTRL[DBG_STOP_TRG], enables a hardware trigger such that, when it asserts, acquisition terminates immediately, in Packet RAM debug mode, as indicated in the following table:

DBG_STOP_TRG[3:0]	HARDWARE TRIGGER
0	no trigger
1	FSK PHY: gfsk_preamble_found
2	FSK PHY: aa_sfd_matched
3	Reserved
4	Reserved
5	RXDIG: agc_dcoc_gain_chg
6	TSM: rx_dig_en
7	TSM: tsm_spare3_en
8	TSM: pll unlock
9	BLE: crc_err_count_incr (increment DTM RX bad packet counter)
10	CRC FAIL: GEN_FSK only, selected by PROTOCOL[3:0]
11	HEADER FAIL: GEN_FSK only
12-15	Reserved

Note: For DBG_STOP_TRIG=11 (HEADER FAIL), register GEN_WHITEN_SZ_THR[REC_BAD_PKT] must be set to 0 (default) to cause an RX recycle on a header violation.

45.3.8.4 Clocks

The module xcvr_dma includes the Transceiver DMA engine and the Packet RAM Debug engine. Each has its own clock:

1. ipg_clk_dma serves the Transceiver DMA hardware
2. ipg_clk_ram_dbg serves the Packet RAM debug hardware

The **ipg_clk_dma** is a gated 32mhz (or 26mhz) clock from the reference oscillator. This clock is activated when DMA_CTRL[DMA_EN]=1, or when DMA_CTRL[DMA_PAGE] is set to a non-zero value, and deactivated when set to 0. Several other conditions allow this clock to free-run; for example, if the DMA request ipd_req_radio_rx is asserted, or if the DMA_TIMED_OUT status bit is asserted awaiting software clearing, **ipg_clk_dma** will free-run until those conditions are no longer true

The **ipg_clk_ram_dbg** is a gated 32mhz (or 26mhz) clock from the reference oscillator. This clock is activated when PACKET_RAM_CTRL[DBG_EN]=1, or when PACKET_RAM_CTRL[DBG_PAGE] is set to a non-zero value.

These 2 clocks are synchronous to each other. All inputs to xcvr_dma module, including both those destined for the Transceiver DMA engine and the Packet RAM debug engine, are also synchronized to the reference oscillator domain. There are no clock domain crossings in this module.

The DMA DarkBlue line signals can cross clock domain boundaries, when SINGLE_REQ_MODE=1. The clock domain crossings are from the reference oscillator domain to the MCU clock domain for ipd_req_radio_rx, and vice versa for ipd_done_radio_rx. Synchronization, both ways, is accomplished outside the radio, in an SoC level block called radio_dma_sync_glue. These signals cross no clock domains inside the radio itself.

The DMA SkyBlue line signal ips_xfr_wait_xcvr can cross clock domain boundaries, when SINGLE_REQ_MODE=1. The clock domain crossing is from the reference oscillator domain to the MCU clock domain. Synchronization is accomplished in the XCVR IPS bus gasket.

There are no special requirements for these clocks, and no anticipated critical paths associated with them.

45.3.8.5 Reset

The xcvr_dma module 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.

45.3.8.6 Interrupts

There are no interrupts associated with either Transceiver DMA mode or Packet RAM Debug mode.

45.3.9 Transceiver Memory Map and Register Definition

Transceiver registers are detailed in following section.

NOTE

The register names XCVR_CTRL and XCVR_MISC are used interchangeably in select instances to refer to the XCVR_MISC module.

45.3.9.1 XCVR_MISC register descriptions

45.3.9.1.1 XCVR_CTRL_ADDR Memory map

XCVR_MISC base address: 4005_C280h

Offset	Register	Width (In bits)	Access	Reset value
0h	TRANSCEIVER CONTROL (XCVR_CTRL)	32	RW	0010_1000h
4h	TRANSCEIVER STATUS (XCVR_STATUS)	32	W1C	See description.
8h	BLE ARBITRATION CONTROL (BLE_ARB_CTRL)	32	RW	0000_0000h
Ch	OVERWRITE VERSION (OVERWRITE_VER)	32	RW	0000_0000h
10h	DIGITAL TEST MUX CONTROL (DTEST_CTRL)	32	RW	0000_0000h
14h	TRANSCEIVER DMA CONTROL (DMA_CTRL)	32	RW	0000_0300h
18h	TRANSCEIVER DMA DATA (DMA_DATA)	32	RO	0000_0000h
1Ch	PACKET RAM CONTROL (PACKET_RAM_CTRL)	32	RW	0000_0000h
20h	PACKET RAM DEBUG RAM STOP ADDRESS (RAM_STOP_ADDR)	32	RO	0000_0000h
24h	FAD CONTROL (FAD_CTRL)	32	RW	0000_F080h

Table continues on the next page...

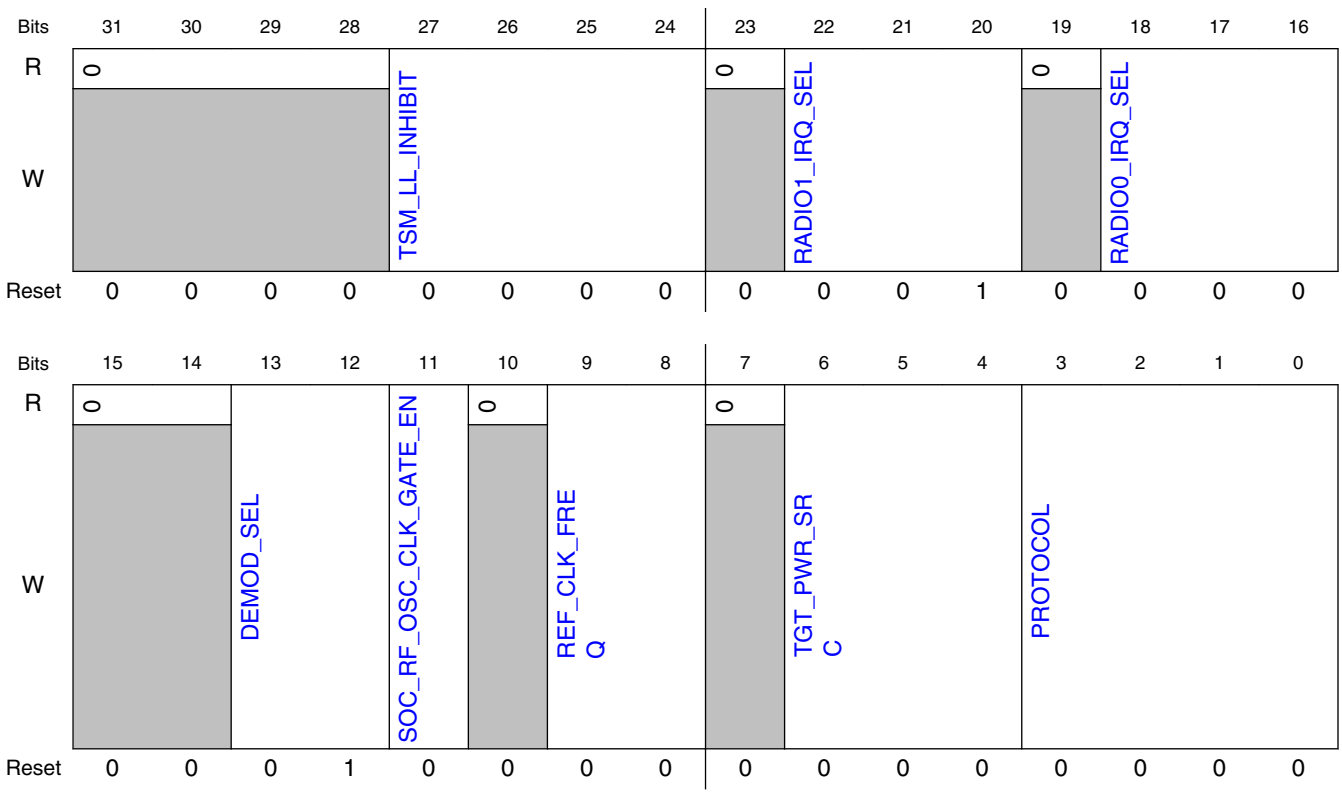
Offset	Register	Width (In bits)	Access	Reset value
2Ch	COEXISTENCE CONTROL (COEX_CTRL)	32	RW	0000_0000h
30h	CRC/WHITENER CONFIG REGISTER (CRCW_CFG)	32	RW	0000_0000h
34h	CRC ERROR CORRECTION MASK (CRC_EC_MASK)	32	RO	0000_0000h
38h	CRC RESULT (CRC_RES_OUT)	32	RO	0000_0000h
3Ch	CRC/WHITENER CONFIG 2 REGISTER (CRCW_CFG2)	32	RW	0000_0000h

45.3.9.1.2 TRANSCEIVER CONTROL (XCVR_CTRL)

45.3.9.1.2.1 Offset

Register	Offset
XCVR_CTRL	0h

45.3.9.1.2.2 Diagram



45.3.9.1.2.3 Fields

Field	Function
31-28 —	Reserved
27-24 TSM_LL_INHIBIT	TSM Per-Link-Layer Inhibit Nominally, any protocol engine can request TSM to start a TX or RX sequence at any time. In a multiprotocol setting, individual protocol engines can (optionally) be selectively blocked from accessing TSM by setting one of the following bits of TSM_LL_INHIBIT[3:0]: **xxx1**: BLE commands to TSM are inhibited **xxx0**: BLE commands to TSM are permitted **1xxx**: GENERIC_FSK commands to TSM are inhibited **0xxx**: GENERIC_FSK commands to TSM are permitted **NOTE:** TSM_LL_INHIBIT[1] and TSM_LL_INHIBIT[2] currently have no functionality
23 —	Reserved
22-20 RADIO1_IRQ_SEL	RADIO1_IRQ_SEL Assigns Radio #1 Interrupt (ipi_int_radio1) to a Protocol Engine 000b - Assign Radio #1 Interrupt to BLE 001b - Radio #1 Interrupt unassigned 010b - Radio #1 Interrupt unassigned 011b - Assign Radio #1 Interrupt to GENERIC_FSK 100b - Radio #1 Interrupt unassigned 101b - Radio #1 Interrupt unassigned 110b - Radio #1 Interrupt unassigned 111b - Radio #1 Interrupt unassigned
19 —	Reserved
18-16 RADIO0_IRQ_SEL	RADIO0_IRQ_SEL Assigns Radio #0 Interrupt (ipi_int_radio0) to a Protocol Engine 000b - Assign Radio #0 Interrupt to BLE 001b - Radio #0 Interrupt unassigned 010b - Radio #0 Interrupt unassigned 011b - Assign Radio #0 Interrupt to GENERIC_FSK 100b - Radio #0 Interrupt unassigned 101b - Radio #0 Interrupt unassigned 110b - Radio #0 Interrupt unassigned 111b - Radio #0 Interrupt unassigned
15-14 —	Reserved.
13-12 DEMOD_SEL	Demodulator Selector This bit selects the demodulator used during reception 00b - No demodulator selected 01b - Use NXP Multi-standard PHY demodulator 10b - Reserved 11b - Reserved
11	SOC_RF_OSC_CLK_GATE_EN

Table continues on the next page...

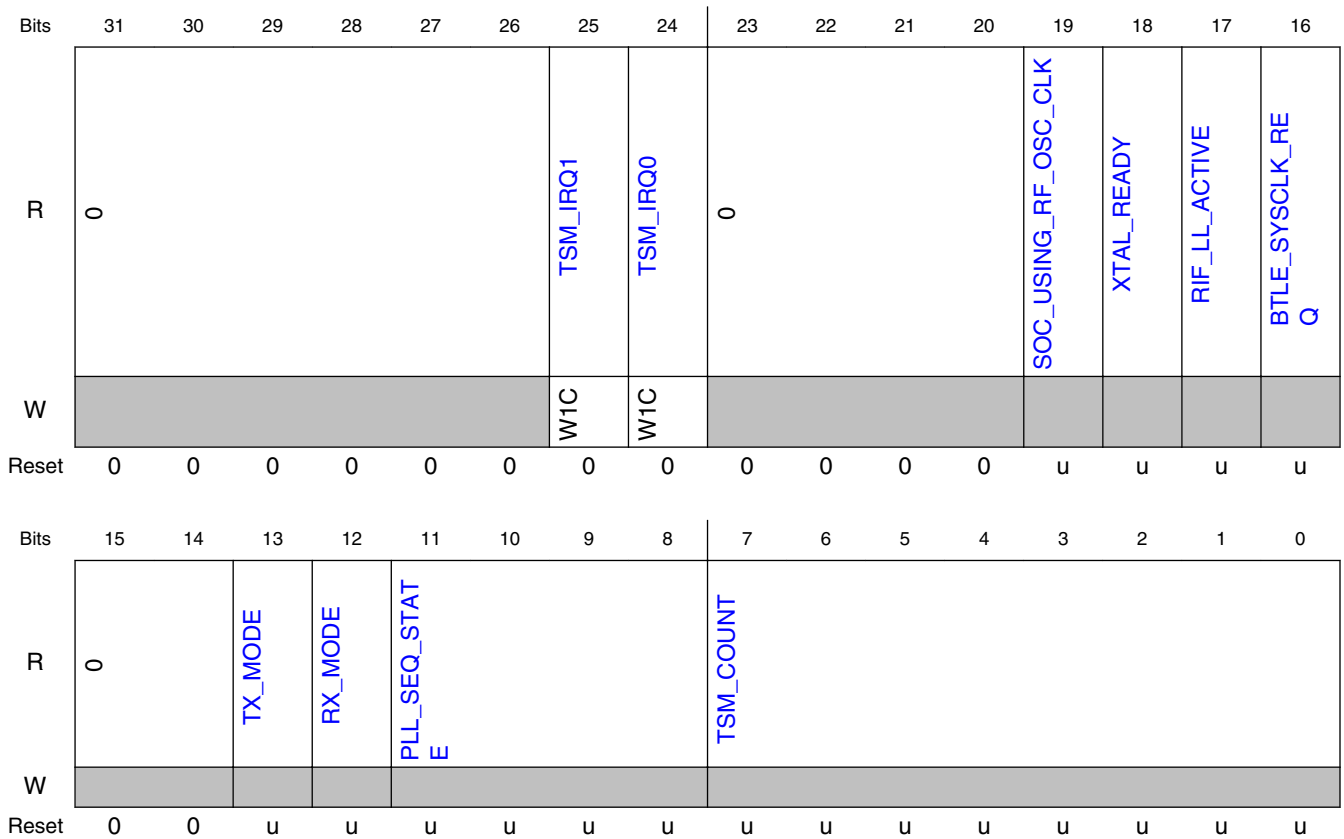
Field	Function
SOC_RF_OSC_CLK_GATE_EN	Enable 3V version of RF OSC Clock for use by the SoC
10 —	Reserved
9-8 REF_CLK_FREQ	Radio Reference Clock Frequency This register selects the Reference Clock Frequency for the Radio. 00b - 32 MHz 01b - 26 MHz 10b - Reserved 11b - Reserved
7 —	Reserved
6-4 TGT_PWR_SRC	Target Power Source For determining transmit power, the TGT_PWR_SRC[2:0] bits control target power selection, according to the following table. 0: TARGET POWER SOURCE is PA_POWER[5:0] register (XCVR space) 1: TARGET POWER SOURCE is BTLE Link Layer 2: Reserved 3: Reserved 4: TARGET POWER SOURCE is GENERIC_FSK Link Layer (TX_POWER[5:0] register in GENERIC_FSK space) 5: Reserved 6: Reserved 7: TARGET POWER SOURCE is determined by the PROTOCOL[3:0] bits
3-0 PROTOCOL	Radio Protocol Selection This register selects the Radio Communication Protocol. 0000b - BLE 0001b - BLE in MBAN 0010b - BLE overlap MBAN 0011b - Reserved 0100b - Reserved 0101b - Reserved 0110b - Radio Channels 0-127 selectable, FSK 0111b - Radio Channels 0-127 selectable, GFSK 1000b - Generic GFSK, with Gaussian Filter 1001b - Generic MSK, O-QPSK encoding 1010b - Generic FSK, direct +/- Fdev FSK

45.3.9.1.3 TRANSCEIVER STATUS (XCVR_STATUS)

45.3.9.1.3.1 Offset

Register	Offset
XCVR_STATUS	4h

45.3.9.1.3.2 Diagram



45.3.9.1.3.3 Fields

Field	Function
31-26 —	Reserved
25 TSM_IRQ1	TSM Interrupt #1 0b - TSM Interrupt #1 is not asserted. 1b - TSM Interrupt #1 is asserted. Write '1' to this bit to clear it.
24 TSM_IRQ0	TSM Interrupt #0 0b - TSM Interrupt #0 is not asserted. 1b - TSM Interrupt #0 is asserted. Write '1' to this bit to clear it.
23-20 —	Reserved
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. 0b - Indicates that the RF Oscillator is disabled or has not completed its warmup.

Table continues on the next page...

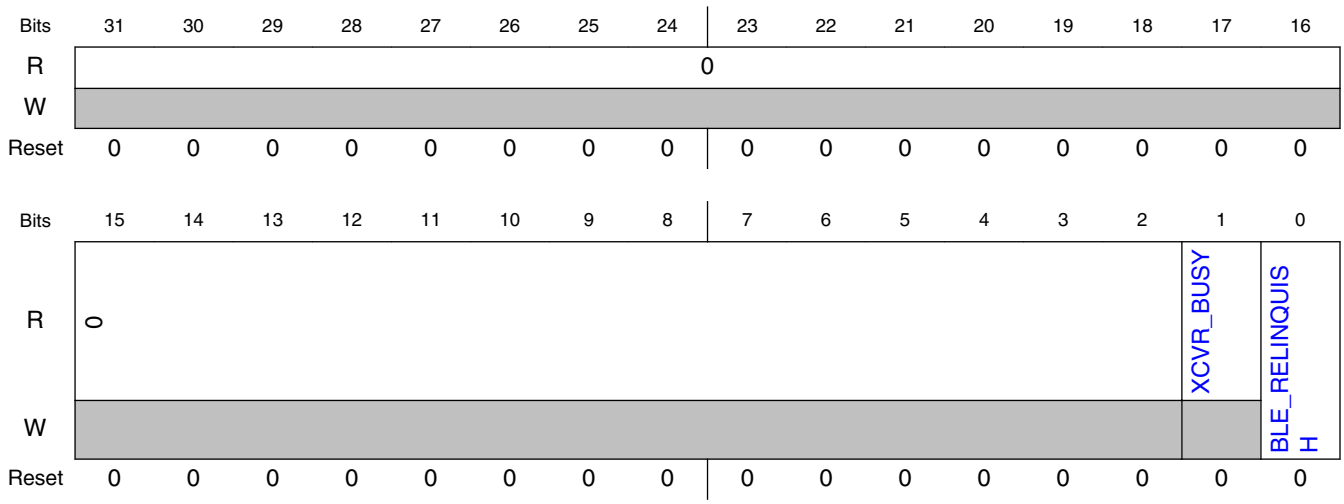
Field	Function
	1b - Indicates that the RF Oscillator has completed its warmup count and is ready for use.
17 RIF_LL_ACTIVE	Link Layer Active Indication Reflects the state of the BTLE “Link Layer Active” status bit. RIF_LL_ACTIVE is to be used by the host as an 'early' indication to prevent host to do any operations while BTLE IP is doing transceiver operations, so as to reduce the peak power and noise.
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
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. 0000b - PLL OFF 0010b - CTUNE 0011b - CTUNE_SETTLE 0110b - HPMCAL1 1000b - HPMCAL1_SETTLE 1010b - HPMCAL2 1100b - HPMCAL2_SETTLE 1111b - PLLREADY
7-0 TSM_COUNT	TSM_COUNT Reflects the instantaneous value of the TSM counter.

45.3.9.1.4 BLE ARBITRATION CONTROL (BLE_ARB_CTRL)

45.3.9.1.4.1 Offset

Register	Offset
BLE_ARB_CTRL	8h

45.3.9.1.4.2 Diagram



45.3.9.1.4.3 Fields

Field	Function
31-2 —	Reserved
1 XCVR_BUSY	Transceiver Busy Status Bit 0b - RF Channel in available (TSM is idle) 1b - RF Channel in use (TSM is busy)
0 BLE_RELINQUISH	BLE Relinquish Control This bit forces the BLE protocol engine to immediately relinquish access to the RF Channel, in favor of another protocol with higher arbitration priority. BLE is denied RF channel access by blocking BLE access to the TSM; when this bit is subsequently cleared, BLE access is restored.

45.3.9.1.5 OVERWRITE VERSION (OVERWRITE_VER)

45.3.9.1.5.1 Offset

Register	Offset
OVERWRITE_VER	Ch

45.3.9.1.5.2 Function

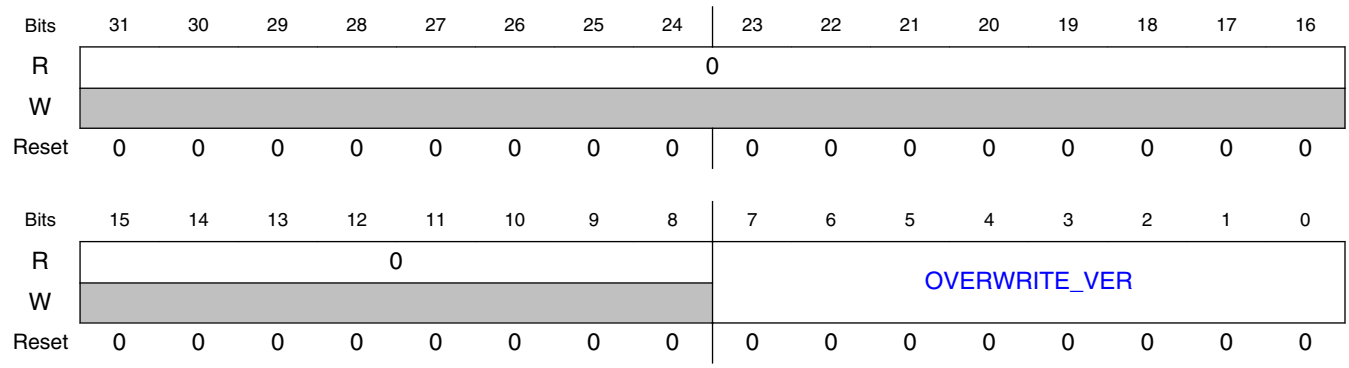
The Overwrite Version allows software to store a version number of trim and calibration values which are used to overwrite the chip default values in the registers. Typically,

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.

45.3.9.1.5.3 Diagram



45.3.9.1.5.4 Fields

Field	Function
31-8 —	Reserved
7-0 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.

45.3.9.1.6 DIGITAL TEST MUX CONTROL (DTEST_CTRL)

45.3.9.1.6.1 Offset

Register	Offset
DTEST_CTRL	10h

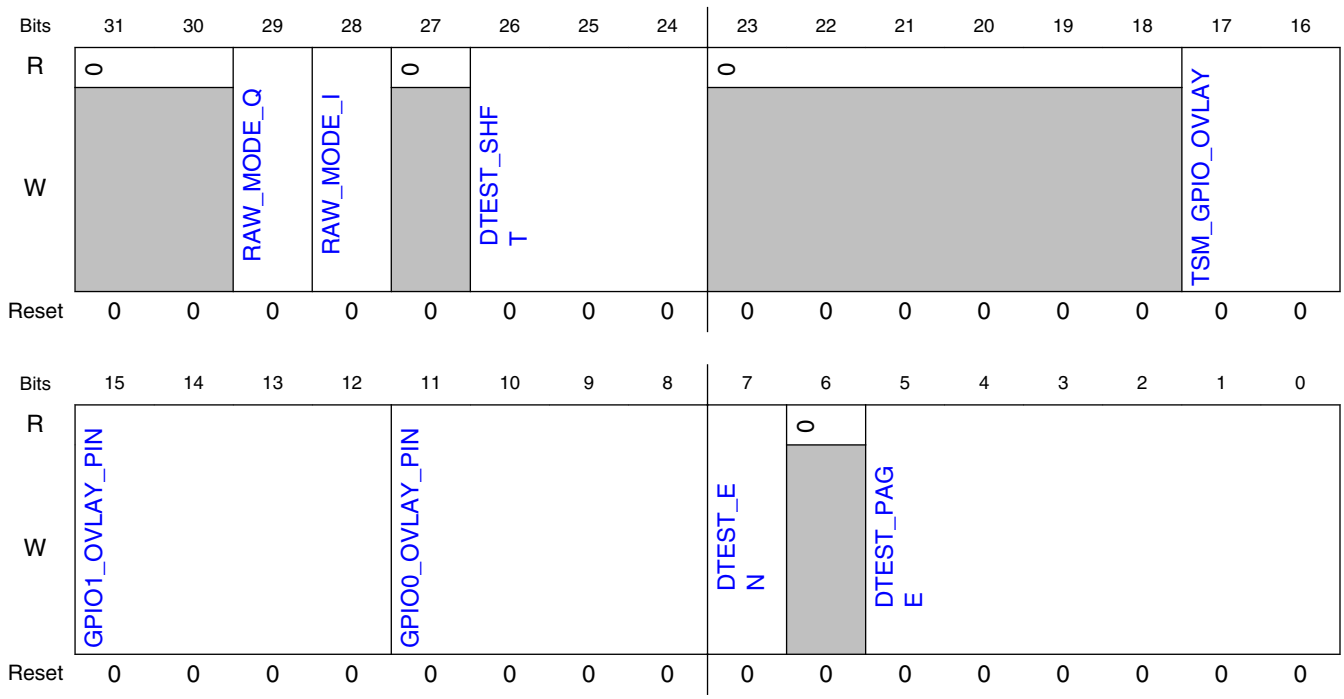
45.3.9.1.6.2 Function

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

45.3.9.1.6.3 Diagram



45.3.9.1.6.4 Fields

Field	Function
31-30 —	Reserved
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	DTEST Raw Mode Enable for I Channel

Table continues on the next page...

Field	Function
RAW_MODE_I	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.
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: PLL_RIPPLE_COUNTER[18:0] 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[18:0], an unsigned value, are shifted by DTEST_SHFT[2:0] according to the following table 0: PLL_RIPPLE_COUNTER[18:5] appear on DTEST[13:0] 1: PLL_RIPPLE_COUNTER[17:4] appear on DTEST[13:0] 2: PLL_RIPPLE_COUNTER[16:3] appear on DTEST[13:0] 3: PLL_RIPPLE_COUNTER[15:2] appear on DTEST[13:0] 4: PLL_RIPPLE_COUNTER[14:1] appear on DTEST[13:0] 5: PLL_RIPPLE_COUNTER[13:0] appear on DTEST[13:0] 6: Reserved 7: Reserved 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: 0: These bits appear on DTEST[13:2] : RX_DIG_IQ[11:0] (no shift) 1: These bits appear on DTEST[13:2] : RX_DIG_IQ[11],RX_DIG_IQ[11:1] (right shift by 1) 2: These bits appear on DTEST[13:2] : RX_DIG_IQ[11],RX_DIG_IQ[11],RX_DIG_IQ[11:2] (right shift by 2) 3: These bits appear on DTEST[13:2] : RX_DIG_IQ[11:0] (no shift)
23-18 ——	Reserved
17-16 TSM_GPIO_OVLAY	TSM GPIO Overlay Pin Control 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. Similarly, the TSM-controlled output GPIO1_TRIG_EN can be routed to any DTEST pin, 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[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[0] = 0 and TSM_GPIO_OVLAY[1] = 0, there is no overlay, and the DTEST Page Table dictates the node that appears on each DTEST pin. 00b - there is no overlay, and the DTEST Page Table dictates the node that appears on each DTEST pin.

Table continues on the next page...

Field	Function
	01b - the register GPIO0_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO0_TRIG_EN will appear. 10b - the register GPIO1_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO1_TRIG_EN will appear. 11b - the register GPIO0_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO0_TRIG_EN will appear, and the register GPIO1_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO1_TRIG_EN will appear.
15-12 GPIO1_OVLAY_PIN	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, 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 0b - Disables DTEST. The DTEST pins assume their mission function. 1b - Enables DTEST. The contents of the selected page (DTEST_PAGE) will appear on the DTEST output pins.
6 —	Reserved
5-0 DTEST_PAGE	DTEST Page Selector DTEST Page signal assignments 000000b - PLLFREQCAL 000001b - PLLBESTDIFF 000010b - PLLRIPPLE 000011b - PLLHPMCAL 000100b - PLLVCOMOD 000101b - PLLUNLOCK 000110b - PLLCYCSLIP 000111b - PLLCHAN 001000b - TXWARMUP 001001b - TXPOWER 001010b - TXFREQWORD 001011b - RXWARMUP 001100b - RXADC 001101b - RXDMA 001110b - RXDIGIQ 001111b - RXDMA2 010000b - RXINPH 010001b - RSSI0 010010b - RSSI1 010011b - AGC0 010100b - AGC1 010101b - DCOC0 010110b - DCOC1 010111b - DCOC2 011000b - DCOC3

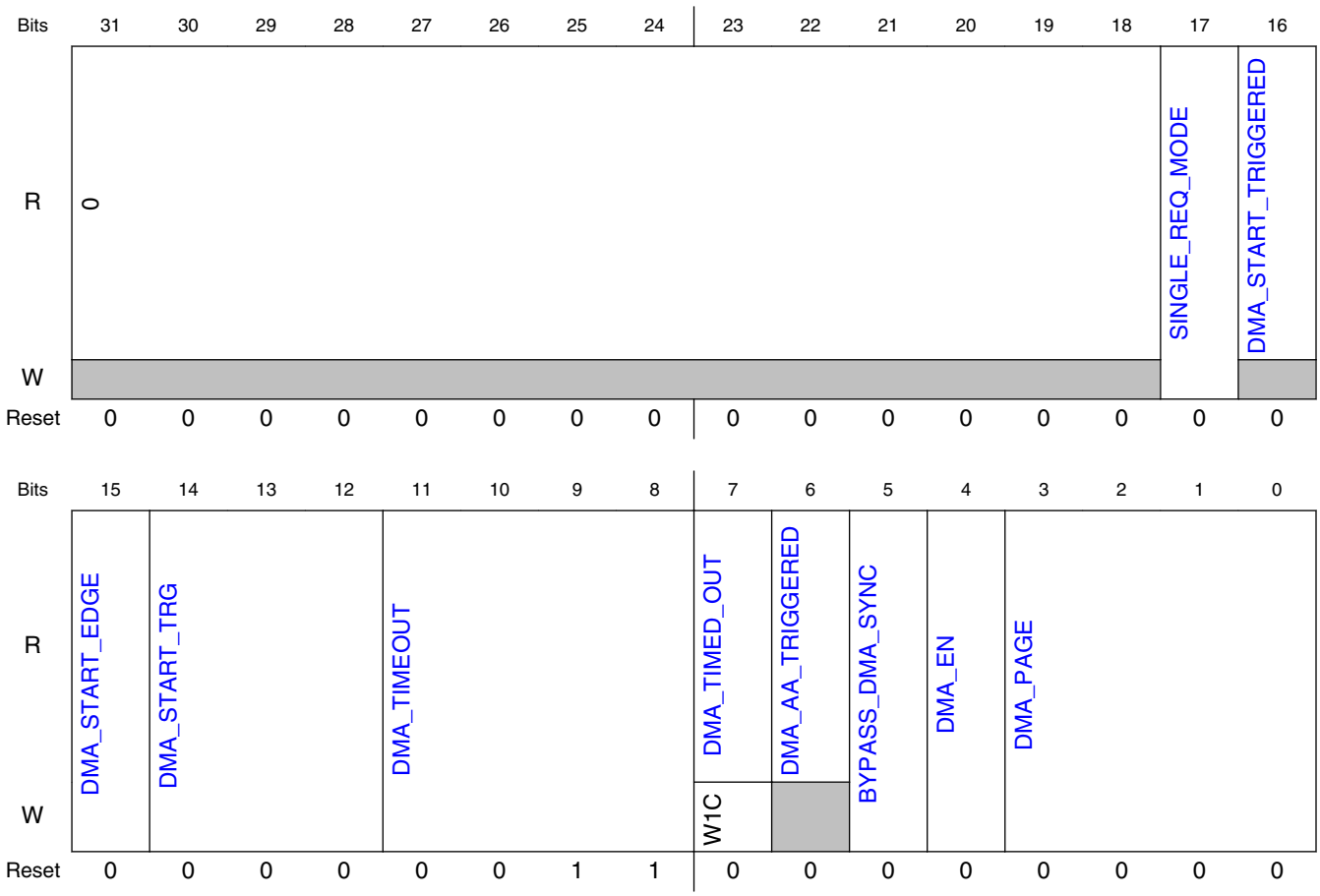
Field	Function
	011001b - TSM 011010b - MTTSMCAL 011011b - MTADV 011100b - MTINIT 011101b - MTSCAN 011110b - MTCONN 011111b - MTDTM 100000b - MTADVXCV 100001b - MTCONXCV 100010b - MTDTM2 100011b - DSM 100100b - PHY_FSK_STATE 100101b - PHY_CFO_EST_PD 100110b - PHY_CFO_EST_PD2 100111b - PHY_EARLY_LATE 101000b - PHY_FSK_DEMOD 101001b - PHY_AA_SEARCH 101010b - PHY_DATA_OUT 101011b - PHY_SAMP_TIME 101100b - CCA_ED_LQI 101101b - CCA_ED_LQI2 101110b - Reserved 101111b - Reserved 110000b - Reserved 110001b - Reserved 110010b - Reserved 110011b - Reserved 110100b - Reserved 110101b - Reserved 110110b - Reserved 110111b - Reserved 111000b - Reserved 111001b - Reserved 111010b - RCCAL 111011b - AUXPLLFCAL 111100b - GENFSKTX 111101b - GENFSKRX 111110b - GENFSKSTATE 111111b - GENFILTER

45.3.9.1.7 TRANSCEIVER DMA CONTROL (DMA_CTRL)

45.3.9.1.7.1 Offset

Register	Offset
DMA_CTRL	14h

45.3.9.1.7.2 Diagram



45.3.9.1.7.3 Fields

Field	Function
31-18 —	Reserved
17 SINGLE_REQ_MODE	DMA Single Request Mode Configures the transceiver to transfer the entire block of DMA data with only a single initial request (ipd_req_radio_rx). The DMA2 controller must be configured accordingly. In this mode, the transceiver will use **ips_xfr_wait** to pace the individual transactions. Single Request Mode should not be used with DMA Pages 11, 12, or 13, because the data rate is too low and therefore ips_xfr_wait would remain assserted for excessively long periods. 0b - Disable Single Request Mode. The transceiver will assert ipd_req_radio_rx whenever it has a new sample ready for transfer. 1b - Enable Single Request Mode. A single initial request by the transceiver will transfer the entire DMA block of data
16 DMA_START_TRIGGERED	DMA Start Trigger Occurred This read-only status bit becomes set, when, during a DMA session (DMA_PAGE > 0), the trigger source selected by DMA_START_TRG[2:0] occurs, with the edge sensitivity selected by DMA_START_EDGE. To clear this bit, set DMA_PAGE=0.

Table continues on the next page...

Field	Function
15 DMA_START_EDGE	DMA Start Trigger Edge Selector DMA_START_EDGE selects the edge sensitivity (rising or falling) of the selected start-trigger. 0b - Trigger fires on a rising edge of the selected trigger source 1b - Trigger fires on a falling edge of the selected trigger source
14-12 DMA_START_TRG	DMA Start Trigger Selector The DMA Start Trigger, if desired, can be selected from the sources in the following table: 0: no trigger 1: FSK PHY: gfsk_preamble_found 2: FSK PHY: aa_sfd_matched 3: Reserved 4: Reserved 5: RXDIG: agc_dcoc_gain_chg 6: TSM: rx_dig_en 7: TSM: tsm_spare2_en
11-8 DMA_TIMEOUT	DMA Timeout In DMA Single Request Mode, adverse consequences may result if the transceiver's ips_xfr_wait signal is asserted for an excessively long period of time, which could occur due to mis-programming of the DMA2 controller, and/or the transceiver itself. DMA Timeout forces the transceiver's ips_xfr_wait low after *N* microseconds and sets the **DMA_TIMED_OUT** status bit, where *N*=DMA_TIMEOUT. DMA_TIMEOUT is only relavent when SINGLE_REQ_MODE=1, otherwise the transceiver does not assert ips_xfr_wait.
7 DMA_TIMED_OUT	DMA Transfer Timed Out Status bit indicates that the transceiver DMA, while operating in Single Request Mode, asserted ips_xfr_wait for a period in excess of the programmed **DMA_TIMEOUT** setting, resulting in a timeout condition where the transceiver has forced ips_xfr_wait low. After a timeout, before resuming DMA operations, set DMA_PAGE=0 to disable DMA, and write 1 to this bit to clear it. 0b - A DMA timeout has not occurred 1b - A DMA timeout has occurred in Single Request Mode since the last time this bit was cleared
6 DMA_AA_TRIGGERED	DMA Access Address triggered This status bit becomes set when the PHY indicates that a Network Address match has occurred. In some cases it may be desirable to delay the start of DMA transfers until such a match occurs; in such cases, software can poll this bit, and once it becomes set, immediately select the desired DMA_PAGE to initiate DMA transfers. This bit is intended for use with DMA_PAGE=13, but is available on all pages. Its usage is optional.
5 BYPASS_DMA_SYNC	Bypass External DMA Synchronization When using transceiver DMA with SINGLE_REQ_MODE=0, synchronization external to the transceiver must be bypassed in order to minimize bus access latency. Using DMA in this mode also requires that the MCU and transceiver both are configured to operate in the RF Oscillator clock domain. 0b - Don't Bypass External Synchronization. Use this setting if SINGLE_REQ_MODE=1. 1b - Bypass External Synchronization. This setting is mandatory if SINGLE_REQ_MODE=0.
4 DMA_EN	DMA Enable Setting DMA_EN=1 enables clock gating to the Transciever DMA engine. This bit should be set prior to the start of a DMA session, and remain set for the duration of the session.
3-0 DMA_PAGE	Transceiver DMA Page Selector Selects the page of receiver data for storage to system memory when using Transceiver DMA Debug Mode. Setting this register to a non-zero value enables the DMA interface logic. Setting this register to zero disables the interface. **Note:** DMA_PAGE should be set to a non-zero value to begin an acquisition, only after the start trigger has been programmed (DMA_START_TRG), if a trigger is desired. The available DMA pages are listed below. 0000b - DMA Idle 0001b - RX_DIG I and Q

Field	Function
	0010b - RX_DIG I Only 0011b - RX_DIG Q Only 0100b - RAW ADC I and Q 0101b - RAW ADC I Only 0110b - RAW ADC Q only 0111b - DC Estimator I and Q 1000b - DC Estimator I Only 1001b - DC Estimator Q only 1010b - RX_DIG Phase Output 1011b - Reserved 1100b - Demodulator Soft Decision 1101b - Demodulator Data Output 1110b - Demodulator CFO Phase Output 1111b - Reserved

45.3.9.1.8 TRANSCEIVER DMA DATA (DMA_DATA)

45.3.9.1.8.1 Offset

Register	Offset
DMA_DATA	18h

45.3.9.1.8.2 Diagram

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

45.3.9.1.8.3 Fields

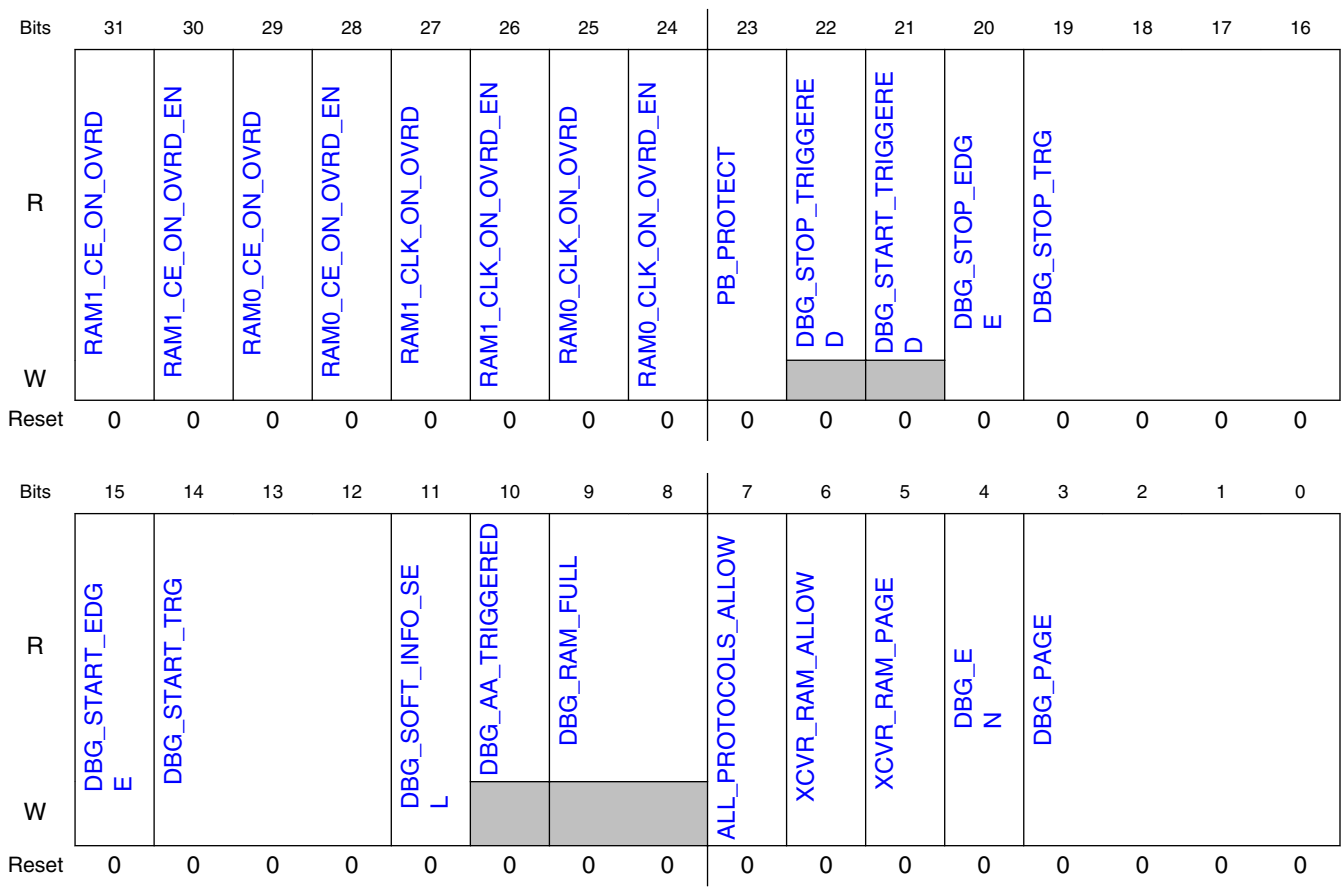
Field	Function
31-0 DMA_DATA	DMA Data Register This register is the singular address location at which the SoC DMA controller accesses samples from the transceiver for transfer to system memory. During DMA operation, the contents of this register depend on the DMA Page selected. The DMA Data register is intended for DMA purposes only, not Host IPS accesses, but Host IPS bus access to this register is not prohibited.

45.3.9.1.9 PACKET RAM CONTROL (PACKET_RAM_CTRL)

45.3.9.1.9.1 Offset

Register	Offset
PACKET_RAM_CTRL	1Ch

45.3.9.1.9.2 Diagram



45.3.9.1.9.3 Fields

Field	Function
31 RAM1_CE_ON_OVRD	Override value for RAM1 CE (Chip Enable) When RAM1_CE_ON_OVRD_EN=1, this value overrides the mission mode state of the RAM1 CE. This bit is ignored when RAM1_CE_ON_OVRD_EN=0.

Table continues on the next page...

Field	Function
30 RAM1_CE_ON_OVRD_EN	Override control for RAM1 CE (Chip Enable) 0b - Normal operation. 1b - Use the state of RAM1_CE_ON_OVRD to override the RAM1 CE.
29 RAM0_CE_ON_OVRD	Override value for RAM0 CE (Chip Enable) When RAM0_CE_ON_OVRD_EN=1, this value overrides the mission mode state of the RAM0 CE. This bit is ignored when RAM0_CE_ON_OVRD_EN=0.
28 RAM0_CE_ON_OVRD_EN	Override control for RAM0 CE (Chip Enable) 0b - Normal operation. 1b - Use the state of RAM0_CE_ON_OVRD to override the RAM0 CE.
27 RAM1_CLK_ON_OVRD	Override value for RAM1 Clock Gate Enable When RAM1_CLK_ON_OVRD_EN=1, this value overrides the mission mode state of the RAM1 Clock Gate Enable. This bit is ignored when RAM1_CLK_ON_OVRD_EN=0.
26 RAM1_CLK_ON_OVRD_EN	Override control for RAM1 Clock Gate Enable 0b - Normal operation. 1b - Use the state of RAM1_CLK_ON_OVRD to override the RAM1 Clock Gate Enable.
25 RAM0_CLK_ON_OVRD	Override value for RAM0 Clock Gate Enable When RAM0_CLK_ON_OVRD_EN=1, this value overrides the mission mode state of the RAM0 Clock Gate Enable. This bit is ignored when RAM0_CLK_ON_OVRD_EN=0.
24 RAM0_CLK_ON_OVRD_EN	Override control for RAM0 Clock Gate Enable 0b - Normal operation. 1b - Use the state of RAM0_CLK_ON_OVRD to override the RAM0 Clock Gate Enable.
23 PB_PROTECT	Packet Buffer Protect Protect Packet Buffer contents against overwriting by the next received packet. Applies to all protocols except BLE 0b - Incoming received packets overwrite Packet Buffer RX contents (default) 1b - Incoming received packets are blocked from overwriting Packet Buffer RX contents
22 DBG_STOP_TRIGGERED	Packet RAM Debug Stop Triggered This read-only status bit becomes set, when, during a Packet RAM Debug session (DBG_PAGE > 0), the trigger source selected by DBG_STOP_TRG[3:0] occurs, with the edge sensitivity selected by DBG_STOP_EDGE. To clear this bit, set DBG_PAGE=0.
21 DBG_START_TRIGGERED	Packet RAM Debug Start Triggered This read-only status bit becomes set, when, during a Packet RAM Debug session (DBG_PAGE > 0), the trigger source selected by DBG_START_TRG[2:0] occurs, with the edge sensitivity selected by DBG_START_EDGE. To clear this bit, set DBG_PAGE=0.
20 DBG_STOP_EDGE	Packet RAM Debug Stop Trigger Edge Selector DBG_STOP_EDGE selects the edge sensitivity (rising or falling) of the selected stop-trigger. 0b - Trigger fires on a rising edge of the selected trigger source 1b - Trigger fires on a falling edge of the selected trigger source
19-16 DBG_STOP_TRG	Packet RAM Debug Stop Trigger Selector The Packet RAM Debug Stop Trigger, if desired, can be selected from the sources in the following table: 0: no trigger 1: FSK PHY: gfsk_preamble_found 2: FSK PHY: aa_sfd_matched 3: Reserved 4: Reserved 5: RXDIG: agc_dcoc_gain_chg

Table continues on the next page...

Field	Function
	6: TSM: rx_dig_en 7: TSM: tsm_spare3_en 8: TSM: pll unlock 9: BLE: crc_err_count_incr (increment DTM RX bad packet counter) 10: CRC FAIL: GEN_FSK only 11: HEADER FAIL: GEN_FSK only 12-15: Reserved
15 DBG_START_EDGE	Packet RAM Debug Start Trigger Edge Selector DBG_START_EDGE selects the edge sensitivity (rising or falling) of the selected start-trigger. 0b - Trigger fires on a rising edge of the selected trigger source 1b - Trigger fires on a falling edge of the selected trigger source
14-12 DBG_START_TRG	Packet RAM Debug Start Trigger Selector The Packet RAM Debug Start Trigger, if desired, can be selected from the sources in the following table: 0: no trigger 1: FSK PHY: gfsk_preamble_found 2: FSK PHY: aa_sfd_matched 3: Reserved 4: Reserved 5: RXDIG: agc_dcoc_gain_chg 6: TSM: rx_dig_en 7: TSM: tsm_spare2_en
11 DBG_SOFT_INFO_SEL	Packet RAM Debug PHY Soft Info Output Selector When acquiring data in Packet RAM Debug Mode with the DBG_PAGE=DMDSOFT page, this bit selects the signal from the PHY used to capture the soft decision data. **NOTE:** When the soft bits are captured without regard to aa_sfd_matched, the bits will be sampled at a higher rate before Access Address detection. Prior to the assertion of aa_sfd_matched, the soft bits are at OSR 8. Following the assertion of aa_sfd_matched plus 2, the soft bits are at OSR 2. 0b - PHY input cg_vbr_en is used to capture soft decision data 1b - PHY output fsk_demod_bit_valid is used to capture soft decision data
10 DBG_AA_TRIGGERED	Packet Ram Debug Access Address triggered This status bit becomes set when the PHY indicates that a Network Address match has occurred. In some cases it may be desirable to delay the start of Packet RAM debug transfers until such a match occurs; in such cases, software can poll this bit, and once it becomes set, immediately select the desired DBG_PAGE to initiate data transfers to Packet RAM. This bit is intended for use with DBG_PAGE=13, but is available on all pages. Its usage is optional.
9-8 DBG_RAM_FULL	DBG_RAM_FULL[1:0] Status Bits indicating that Packet RAM0 (or RAM1) is full, and the Packet RAM Debug engine has attempted to write another word to that RAM. This, and any subsequent write attempts, will not alter the contents of the RAM, once it has reached capacity. To clear the DBG_RAM_FULL[1:0] bits, the DBG_PAGE field must be set to 0 and then set to a non-zero value (which starts a new capture by the Packet RAM Debug engine) That is, after a capture, the DBG_RAM_FULL[1:0] bits only reset to zero upon the start of the subsequent capture, at which time it is safe to start polling these bits again. **NOTE:** If a Debug Stop Trigger is selected (DBG_STOP_TRG > 0), writes to RAM do not stop when DBG_RAM_FULL bit(s) become set; instead, writes continue after an address wrap-around to 0, and will continue indefinitely until either a stop trigger condition occurs, or the debug session is ended by taking DBG_PAGE=0. The DBG_RAM_FULL[1:0] remain a valid indicator that the RAM(s) has reached full status at least once during the debug session. 00b - Neither Packet RAM0 nor RAM1 is full 1xb - Packet RAM1 has been filled to capacity. x1b - Packet RAM0 has been filled to capacity.

Table continues on the next page...

Field	Function
7 ALL_PROTOCOLS_ALLOW	Allow IPS bus access to Packet RAM for any protocol at any time. Each supported protocol has an associated IPS bus, with which it can access protocol-specific registers, as well as Packet RAM. Normally IPS bus access to the Packet RAM is restricted to the protocol currently selected by the XCVR_CTRL[PROTOCOL] register, and access attempts by other protocols will result in an assert of **ips_xfr_err** on the offending IPS interface. When ALL_PROTOCOLS_ALLOW=1, these inhibits are removed, and any IPS bus can access Packet RAM at any time without any error signalling. 0b - IPS bus access to Packet RAM is restricted to the protocol engine currently selected by XCVR_CTRL[PROTOCOL]. 1b - All IPS bus access to Packet RAM permitted, regardless of XCVR_CTRL[PROTOCOL] setting
6 XCVR_RAM_ALLOW	Allow Packet RAM Transceiver Access This bit must be set before performing accesses to the Packet RAM using transceiver address space. Transceiver space accesses to Packet RAM are intended for debug purposes only; the individual protocol engines, and the associated IPS busses, have exclusive access to Packet RAM in mission modes. Transceiver space accesses to Packet RAM include direct accesses to RAM at transceiver addresses 0x700 - 0xFFF, as well as Packet RAM Debug Mode. When this bit is set, control of RAM clock gating and RAM chip enables are forced on continuously, taking these controls away from the protocol engines. 0b - Protocol Engines, and associated IPS busses, have exclusive access to Packet RAM (mission mode) 1b - Transceiver-space access to Packet RAM, including Packet RAM debug mode, are allowed
5 XCVR_RAM_PAGE	RAM Page Selector for XCVR Access This bit selects which of the 2 Packet RAM blocks is mapped into XCVR address space starting at XCVR_BASE + 0x700. 0b - RAM0 is mapped into XCVR address space, between XCVR_BASE + 0x700, and XCVR_BASE + 0xFFF 1b - RAM1 is mapped into XCVR address space, between XCVR_BASE + 0x700, and XCVR_BASE + 0xFFF
4 DBG_EN	Packet RAM Debug Mode Enable Setting DBG_EN=1 enables clock gating to the Packet RAM Debug engine. This bit should be set prior to the start of a Debug session, and remain set for the duration of the session.
3-0 DBG_PAGE	Packet RAM Debug Page Selector Selects the page of receiver data for storage to Packet RAM using Transceiver Packet RAM Debug Mode. Setting this register to a non-zero value enables the Packet RAM Debug Mode interface logic. Setting this register to zero disables the interface. **Note:** DBG_PAGE should be set to a non-zero value to begin an acquisition, only after the start and stop triggers have been programmed (DBG_START_TRG and DBG_STOP_TRG), if a trigger is desired. The available RAM Debug pages are listed below. 0000b - Packet RAM Debug Mode Idle 0001b - RX_DIG I and Q 0010b - Reserved 0011b - Reserved 0100b - RAW ADC I and Q 0101b - Reserved 0110b - Reserved 0111b - DC Estimator I and Q 1000b - Reserved 1001b - Reserved 1010b - RX_DIG Phase Output 1011b - Reserved 1100b - Demodulator Soft Decision 1101b - Demodulator Data Output 1110b - Demodulator CFO Phase Output 1111b - Reserved

45.3.9.1.10 PACKET RAM DEBUG RAM STOP ADDRESS (RAM_STOP_ADDR)

45.3.9.1.10.1 Offset

Register	Offset
RAM_STOP_ADDR	20h

45.3.9.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					RAM1_STOP_ADDR										
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	0					RAM0_STOP_ADDR										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.3.9.1.10.3 Fields

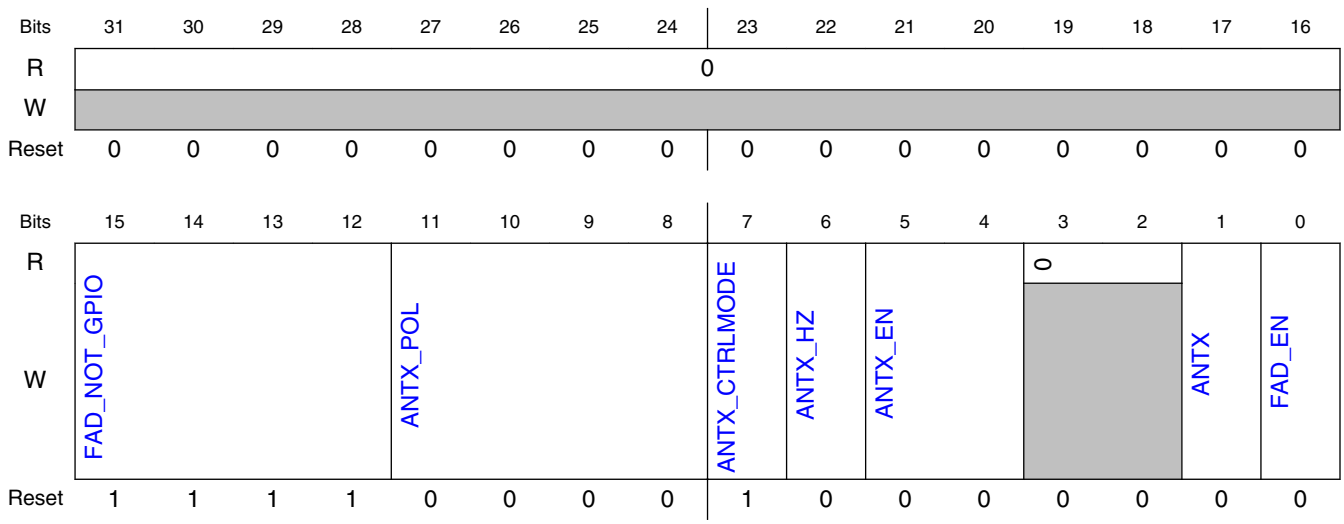
Field	Function
31-27 —	Reserved
26-16 RAM1_STOP_ADDR	RAM1 Stop Address After a Packet RAM Debug Mode acquisition, RAM1_STOP_ADDR represents the last (most recent) RAM1 location filled with sample data before the acquision terminated. The acquisition terminates when a stop-trigger occurs, or due to a manual termination by setting DBG_PAGE=0. RAM1_STOP_ADDR is in units of RAM "words", where each word represents 2 bytes of XCVR RAM address space.
15-11 —	Reserved
10-0 RAM0_STOP_ADDR	RAM0 Stop Address After a Packet RAM Debug Mode acquisition, RAM0_STOP_ADDR represents the last (most recent) RAM0 location filled with sample data before the acquision terminated. The acquisition terminates when a stop-trigger occurs, or due to a manual termination by setting DBG_PAGE=0. RAM0_STOP_ADDR is in units of RAM "words", where each word represents 2 bytes of XCVR RAM address space.

45.3.9.1.11 FAD CONTROL (FAD_CTRL)

45.3.9.1.11.1 Offset

Register	Offset
FAD_CTRL	24h

45.3.9.1.11.2 Diagram



45.3.9.1.11.3 Fields

Field	Function
31-16 —	Reserved
15-12 FAD_NOT_GPIO	FAD versus GPIO Mode Selector **x1xx**: Reserved **x0xx**: The TX_SWITCH pad is controlled directly by the TSM output gpio2_trig_en **1xxx**: Reserved **0xxx**: The RX_SWITCH pad is controlled directly by the TSM output gpio3_trig_en **NOTE:** FAD_NOT_GPIO[0] and FAD_NOT_GPIO[1] currently have no functionality.
11-8 ANTX_POL	FAD Antenna Controls Polarity Control the polarity of the FAD pins: ANTX_POL<0> : This bit currently has no functionality ANTX_POL<1> : This bit currently has no functionality

Table continues on the next page...

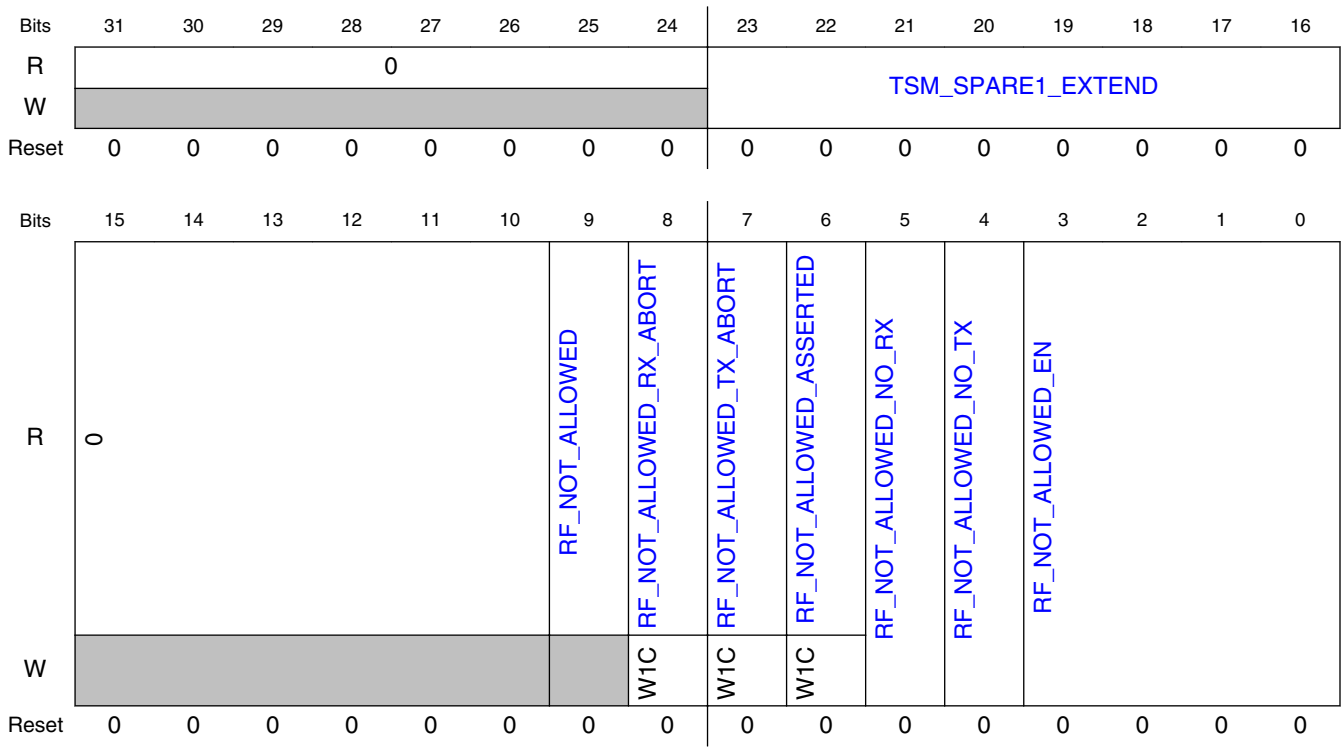
Field	Function
	ANTX_POL<2>=1 : invert the TX_SWITCH output ANTX_POL<3>=1 : invert the RX_SWITCH output
7 ANTX_CTRLMODE	Antenna Diversity Control Mode When ANTX_CTRLMODE=1 (dual mode): TX_SWITCH=GPIO2_TRIG_EN RX_SWITCH=GPIO3_TRIG_EN When ANTX_CTRLMODE=0 (single mode): TX_SWITCH=GPIO2_TRIG_EN RX_SWITCH=(GPIO3_TRIG_EN 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
6 ANTX_HZ	This bit currently has no functionality
5-4 ANTX_EN	These bits currently have no functionality
3-2 —	Reserved.
1 ANTX	This bit currently has no functionality
0 FAD_EN	This bit currently has no functionality

45.3.9.1.12 COEXISTENCE CONTROL (COEX_CTRL)

45.3.9.1.12.1 Offset

Register	Offset
COEX_CTRL	2Ch

45.3.9.1.12.2 Diagram



45.3.9.1.12.3 Fields

Field	Function
31-24 —	Reserved
23-16 TSM_SPARE1_EXTEND	TSM_SPARE1_EX Extension Duration When TSM output TSM_SPARE1_EN is enabled to assert during any TX or RX sequence and TSM_SPARE1_EXTEND[7:0] > 0, this register specifies the number of microseconds for which TSM_SPARE1_EN is to remain asserted beyond the end of the TSM warmdown. This extension is intended to close any gap in the coexistence output RF_ACTIVE that may occur between consecutive TX/RX or RX/TX TSM operations.
15-10 —	Reserved
9 RF_NOT_ALLOWED	RF_NOT_ALLOWED Reflects the instantaneous state of the RF_NOT_ALLOWED pin, synchronized into the RF OSC clock domain.
8 RF_NOT_ALLOWED_RX_ABORT	RF_NOT_ALLOWED_RX_ABORT 0b - A RX abort due to assertion on RF_NOT_ALLOWED has not occurred 1b - A RX abort due to assertion on RF_NOT_ALLOWED has occurred since the last time this bit was cleared
7	RF_NOT_ALLOWED_TX_ABORT 0b - A TX abort due to assertion on RF_NOT_ALLOWED has not occurred

Table continues on the next page...

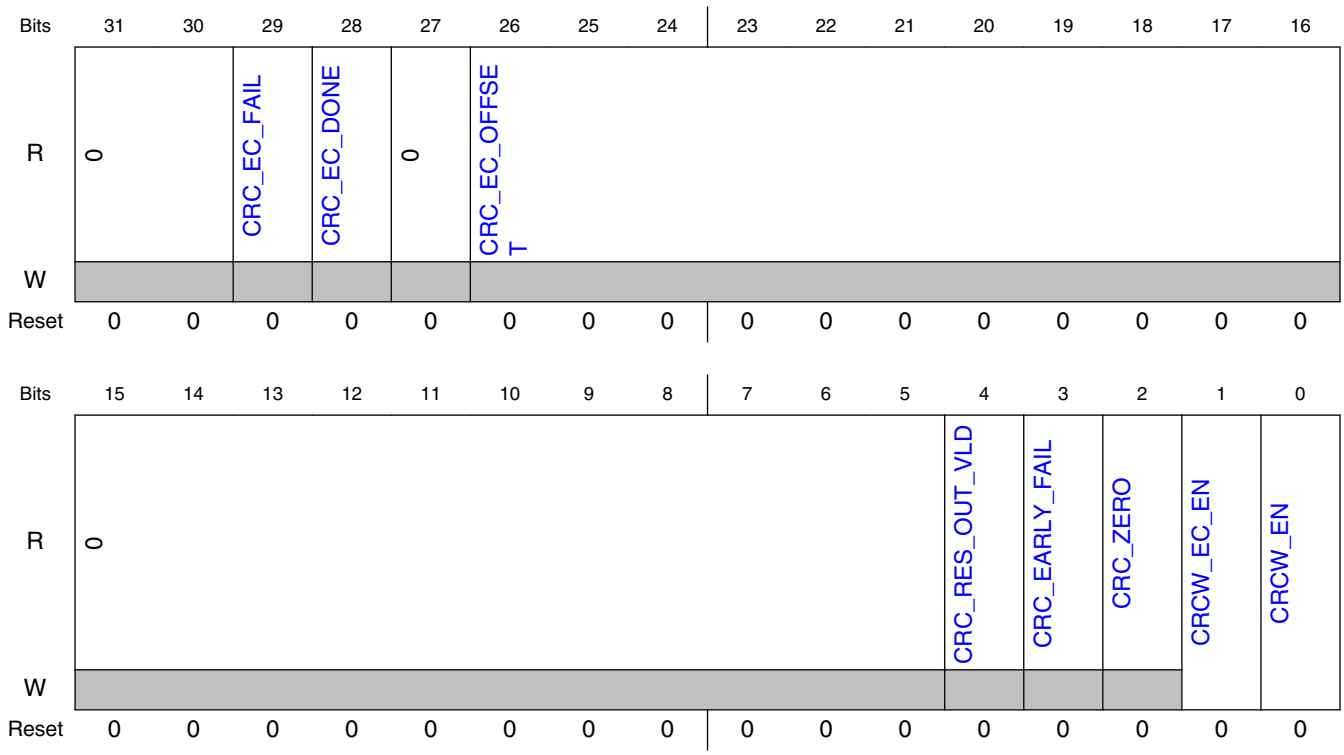
Field	Function
RF_NOT_ALLOWED_TX_ABORT	1b - A TX abort due to assertion on RF_NOT_ALLOWED has occurred since the last time this bit was cleared
6 RF_NOT_ALLOWED_ASSERTED	RF_NOT_ALLOWED_ASSERTED 0b - Assertion on RF_NOT_ALLOWED has not occurred 1b - Assertion on RF_NOT_ALLOWED has occurred since the last time this bit was cleared
5 RF_NOT_ALLOWED_NO_RX	RF_NOT_ALLOWED_NO_RX 0b - Assertion on RF_NOT_ALLOWED has no effect on RX 1b - Assertion on RF_NOT_ALLOWED can abort RX
4 RF_NOT_ALLOWED_NO_TX	RF_NOT_ALLOWED_NO_TX 0b - Assertion on RF_NOT_ALLOWED has no effect on TX 1b - Assertion on RF_NOT_ALLOWED can abort TX
3-0 RF_NOT_ALLOWED_EN	RF_NOT_ALLOWED PER-LINK-LAYER ENABLE The coexistence input RF_NOT_ALLOWED can be enabled to selectively abort TX or RX sequences, with individual enables for each supported protocol, subject also to the state of the RF_NOT_ALLOWED_NO_TX and RF_NOT_ALLOWED_NO_RX control bits, according to the following table: **xxx1**: RF_NOT_ALLOWED assertions are enabled to abort BLE TX and RX sequences **xxx0**: RF_NOT_ALLOWED assertions are not enabled to abort BLE TX and RX sequences **1xxx**: RF_NOT_ALLOWED assertions are enabled to abort GENERIC_FSK TX and RX sequences **0xxx**: RF_NOT_ALLOWED assertions are not enabled to abort GENERIC_FSK TX and RX sequences **NOTE:** RF_NOT_ALLOWED_EN[1] and RF_NOT_ALLOWED_EN[2] currently have no functionality

45.3.9.1.13 CRC/WHITENER CONFIG REGISTER (CRCW_CFG)

45.3.9.1.13.1 Offset

Register	Offset
CRCW_CFG	30h

45.3.9.1.13.2 Diagram



45.3.9.1.13.3 Fields

Field	Function
31-30 —	Reserved
29 CRC_EC_FAIL	CRC error correction fail This signal is cleared when *crc_init* is asserted. It is set after the completion of a CRC error correction calculation that does not find a valid data correction solution or if error correction was disabled and the CRC check found an error.
28 CRC_EC_DONE	CRC error correction done This signal is cleared when *crc_init* is asserted. It is set after the error correction logic completes processing the syndrome. It is immediately set after a packet is received if no error was detected or if error correction is disabled. Otherwise, it can take up to N system clocks after the packet is received to assert, with N = (number of bits used in the CRC calculation, including the CRC value).
27 —	Reserved
26-16 CRC_EC_OFFSET	CRC error correction offset This value provides the byte offset within the data packet to which the CRC error correction mask should be XOR-ed. This value is valid if the *crc_ec_done* signal is asserted and the *crc_ec_fail* signal is not asserted. The offset includes only the bytes used in the CRC calculation, including the CRC value.
15-5	Reserved

Table continues on the next page...

Field	Function
—	
4 CRC_RES_OUT_VLD	CRC result output valid CRC result output valid
3 CRC_EARLY_FAIL	CRC error correction fail This signal is cleared when *crc_init* is asserted. It is set after the completion of a CRC error correction calculation that does not find a valid data correction solution or if error correction was disabled and the CRC check found an error.
2 CRC_ZERO	CRC zero This signal is asserted at any time that the CRC shift register contains a value of zero
1 CRCW_EC_EN	CRC Error Correction Enable CRC Error Correction Enable
0 CRCW_EN	CRC calculation enable Input data bits loaded with this signal asserted are used in the CRC calculation.

45.3.9.1.14 CRC ERROR CORRECTION MASK (CRC_EC_MASK)

45.3.9.1.14.1 Offset

Register	Offset
CRC_EC_MASK	34h

45.3.9.1.14.2 Diagram

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

45.3.9.1.14.3 Fields

Field	Function
31-0	CRC error correction mask

Field	Function
CRC_EC_MASK	This value provides a 32-bit XOR mask that must be applied to the input data packet to correct the burst errors detected by the error correction calculation. This value is valid if the *crc_ec_done* signal is asserted and the *crc_ec_fail* signal is not asserted.

45.3.9.1.15 CRC RESULT (CRC_RES_OUT)

45.3.9.1.15.1 Offset

Register	Offset
CRC_RES_OUT	38h

45.3.9.1.15.2 Diagram

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

45.3.9.1.15.3 Fields

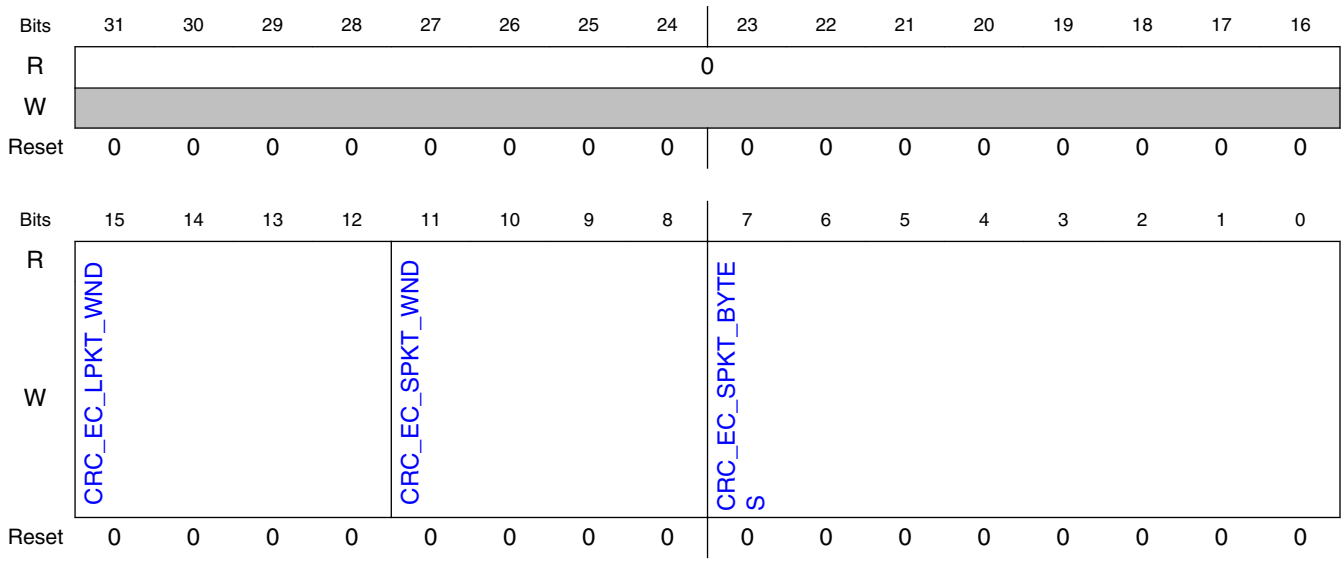
Field	Function
31-0 CRC_RES_OUT	CRC result output This bus provides the instantaneous value of the CRC shift register.

45.3.9.1.16 CRC/WHITENER CONFIG 2 REGISTER (CRCW_CFG2)

45.3.9.1.16.1 Offset

Register	Offset
CRCW_CFG2	3Ch

45.3.9.1.16.2 Diagram



45.3.9.1.16.3 Fields

Field	Function
31-16 —	Reserved
15-12 CRC_EC_LPKT_WND	Error correction long packet burst error window This value determines the largest burst error corrected when long packets are processed. A value of zero will disable long packet error correction. The max value is 15.
11-8 CRC_EC_SPKT_WND	Error correction short packet burst error window This value determines the largest burst error corrected when short packets are processed. A value of zero will disable short packet error correction. The max value is 15.
7-0 CRC_EC_SPKT_BYTES	Error Correction Short Packet Bytes Short packets are defined as those where the number of bytes used in the CRC calculation, including the CRC value, do not exceed this value. The max value is 255.

45.3.9.2 XCVR_PHY register descriptions

Note

To decipher number formats such as ?ufix4en2? or ?sfix8en3?:

General form: [u/s]fix[Nb]e[n][Nx]

Where,

[u/s]	denotes whether the number is signed or not
fix	denotes that the number is fixed point, 2's complement
Nb	indicates total number of bits, including a sign bit if signed
e	indicates that the number is exponentiated like num x 2^(exponent)
n	indicates the exponent is negative
Nx	indicates the value of the exponent

For example :

- ufix4en2(1101) = 1101 x 2^(-2) = 11.01 = 3.25 (decimal)
- sfix8en3(10011011) = -1100101 x 2^(-3) = -1100.101 = -12.625 (decimal)

45.3.9.2.1 XCVR_RX_PHY_ADDR Memory map

XCVR_PHY base address: 4005_C400h

Offset	Register	Width (In bits)	Access	Reset value
0h	PREAMBLE REFERENCE WAVEFORM 0 (PRE_REF0)	32	RW	FDDD_EB39h
4h	PREAMBLE REFERENCE WAVEFORM 1 (PRE_REF1)	32	RW	BEFB_FFFFh
8h	PREAMBLE REFERENCE WAVEFORM 2 (PRE_REF2)	32	RW	0000_CE75h
20h	PHY CONFIGURATION REGISTER 1 (CFG1)	32	RW	1070_CD16h
24h	PHY CONFIGURATION REGISTER 2 (CFG2)	32	RW	0100_0A48h
28h	PHY EARLY/LATE CONFIGURATION REGISTER (EL_CFG)	32	RW	0000_0000h
2Ch	PHY NETWORK ADDRESS FOR BSM (NTW_ADR_BSM)	32	RW	0000_0000h
30h	PHY STATUS REGISTER (STATUS)	32	RO	0000_0000h

45.3.9.2.2 PREAMBLE REFERENCE WAVEFORM 0 (PRE_REF0)

45.3.9.2.2.1 Offset

Register	Offset
PRE_REF0	0h

45.3.9.2.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FSK_PREAMBLE_REF0															
Reset	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FSK_PREAMBLE_REF0															
Reset	1	1	1	0	1	0	1	1	0	0	1	1	1	0	0	1

45.3.9.2.2.3 Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31-0
FSK_PREAMBLE_REF0</td><td>Base preamble reference waveform containing sixteen 5-bit phase values represented in 2's complement notation using 1 sign bit and 4 fractional bits.
Thus, the range of representable values is [-1, 1-1/16] and they correspond to phases in the range [-π, π(1-1/16)].
Preamble reference waveform is an 80-bit vector; FSK_PREAMBLE_REF0 constitutes the lowest (least significant) component. The entire reference waveform is assembled by concatenating the following register values:
Reference Waveform = {FSK_PREAMBLE_REF2[15:0], FSK_PREAMBLE_REF1[31:0], FSK_PREAMBLE_REF0[31:0]
The preamble reference waveforms for various PHY configurations are as follows:
<table>
<tr><th>Generic FSK MODE</th><th>FSK_PREAMBLE-REF0</th><th>FSK_PREAMBLE_REF1</th><th>FSK_PREAMBLE_REF2</th></tr>
<tr><td>BLE (GFSK BT=0.5, h=0.5)</td><td>0x79CDEB39</td><td>0xCE77DEF7</td><td>0x0000CEB7</td></tr>
<tr><td>GFSK BT=0.5, h=0.32</td><td>0xBBDE739B</td><td>0xDEFBDEF7</td><td>0x0000E739</td></tr>
<tr><td>GFSK BT=0.5, h=0.7</td><td>0x37ACE2F7</td><td>0xADF3BDEF</td><td>0x0000BE33</td></tr>
<tr><td>GFSK BT=0.5, h=1.0</td><td>0xF38B5273</td><td>0x8CEF9CE6</td><td>0x00009D2D</td></tr>
<tr><td>GFSK BT=0.3, h=0.5</td><td>0x7BCDEB39</td><td>0xCEF7DEF7</td><td>0x0000CEB7</td></tr>
<tr><td>GFSK BT=0.7, h=0.5</td><td>0x79CDEB39</td><td>0xCE77DEF7</td><td>0x0000CEB7</td></tr>
<tr><td>MSK</td><td>0x79CDEB38</td><td>0xCE77DFF7</td><td>0x0000CEB7</td></tr>
</table>
</td></tr>
</table>

45.3.9.2.3 PREAMBLE REFERENCE WAVEFORM 1 (PRE_REF1)

45.3.9.2.3.1 Offset

Register	Offset
PRE_REF1	4h

45.3.9.2.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FSK_PREAMBLE_REF1															
Reset	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FSK_PREAMBLE_REF1															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

45.3.9.2.3.3 Fields

Field	Function
31-0 FSK_PREAMBLE_REF1	Refer to FSK_PREAMBLE_REF0.

45.3.9.2.4 PREAMBLE REFERENCE WAVEFORM 2 (PRE_REF2)

45.3.9.2.4.1 Offset

Register	Offset
PRE_REF2	8h

45.3.9.2.4.2 Diagram

Bits	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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FSK_PREAMBLE_REF2															
Reset	1	1	0	0	1	1	1	0	0	1	1	1	0	1	0	1

45.3.9.2.4.3 Fields

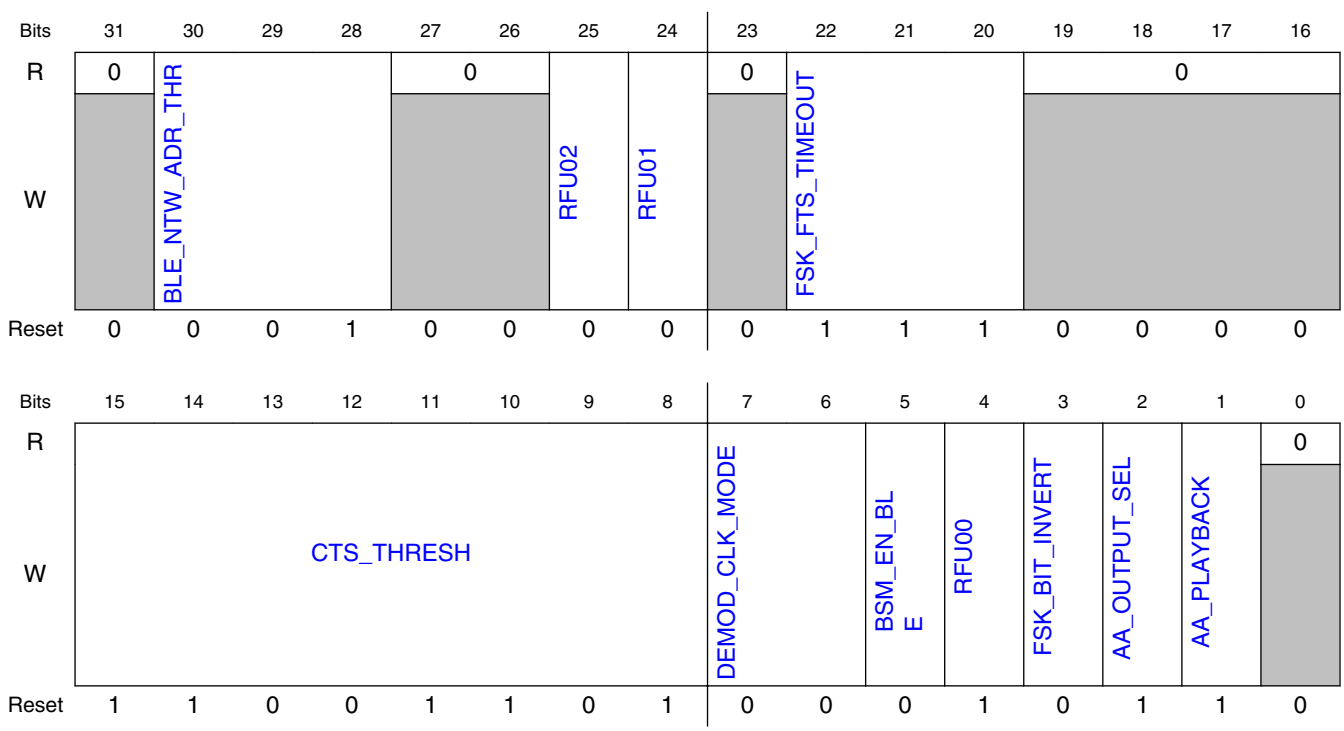
Field	Function
31-16 —	Reserved
15-0 FSK_PREAMBLE_REF2	Refer to FSK_PREAMBLE_REF0.

45.3.9.2.5 PHY CONFIGURATION REGISTER 1 (CFG1)

45.3.9.2.5.1 Offset

Register	Offset
CFG1	20h

45.3.9.2.5.2 Diagram



45.3.9.2.5.3 Fields

Field	Function
31 —	Reserved
30-28 BLE_NTW_ADR_THR	BLE Network Address Match Bit Error Threshold Number of Tolerated bit errors for Access Address correlation in BLE mode
27-26 —	Reserved
25 RFU02	Reserved for future use.
24 RFU01	Reserved for future use.
23 —	Reserved
22-20 FSK_FTS_TIMEOUT	FSK FTS Timeout Number of symbols FTS is allowed to proceed beyond the expected end of the longest AA in FSK 000b - 4 symbols 001b - 5 symbols 010b - 6 symbols 011b - 7 symbols 100b - 8 symbols 101b - 9 symbols 110b - 10 symbols 111b - 11 symbols
19-16 —	Reserved
15-8 CTS_THRESH	CTS Correlation Threshold Coarse Timing Search (CTS) correlation threshold is an unsigned 8-bit fixed-point number that reprsents a range of CTS threshold representable values in [0, ..., 1-1/2^8] = {0000 0000, ..., 1111 1111}. CTS Threshold is a function of the PHY modulation scheme, data rate, and the receiver chain filtering characteristics and is chosen to maximize the preformance. The table below shows the threshold values for an assortment of FSK schemes. They are determined from Simulink model simulations. For more details, please refer to the Multi-PHY chapter in the 2.4GHz Radio 2 ADD.

Modulation Scheme	Data Rate (kbps)	CTS_THRESH value	CTS_THRESH bit representation
BLE (GFSK BT=0.5, h=0.5)	1000	0.7500	1100 0000
ANT (GFSK BT=0.5, h=0.32)	1000	0.8008	1100 1101
GFSK BT=0.5, h=0.7	500	0.7500	1100 0000
GFSK BT=0.3, h=0.5	1000	0.8516	1101 1010
MSK	500	0.8125	1101 0000

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
<table>
<tr><th>Modulation Scheme</th><th>Data Rate (kbps)</th><th>CTS_THRESH value</th><th>CTS_THRESH bit representation</th></tr>
<tr><td>GFSK BT=0.5, h=0.5</td><td>250</td><td>0.8750</td><td>1110 0000</td></tr>
<tr><td>GFSK BT=0.5, h=1.0</td><td>250</td><td>0.6875</td><td>1011 0000</td></tr>
<tr><td>GFSK BT=0.7, h=0.5</td><td>1000</td><td>0.8515</td><td>1101 1010</td></tr>
</table>
</td></tr>
<tr><td>7-6
DEMOD_CLK_MODE</td><td>Demodulator Clock Mode
00b - Normal
01b - Demodulate all samples
10b - Reserved
11b - Reserved</td></tr>
<tr><td>5
BSM_EN_BLE</td><td>BLE Bit Streaming Mode Enable bit
Enable the serialized, received BLE packet bitstream to appear on the BSM pins of the SoC. (See the XCVR BSM Block Guide)
0b - BSM for BLE disabled
1b - BSM for BLE enabled</td></tr>
<tr><td>4
RFU00</td><td>Reserved for future use.</td></tr>
<tr><td>3
FSK_BIT_INVERT</td><td>FSK Bit Invert
Inverts FSK mapping of symbols to bits when asserted.
<table>
<tr><th>FSK_BIT_INVERT</th><th>$E_s \geq 0$</th><th>$E_s < 0$</th></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
</table>
In the table above, E_s is the soft bit output from the symbol demodulator.</td></tr>
<tr><td>2
AA_OUTPUT_SEL</td><td>Access Address Output Select
Selects whether the demodulated AA bit sequence or the matched AA pattern is output.
0b - demodulated
1b - matched</td></tr>
<tr><td>1
AA_PLAYBACK</td><td>Access Address Playback
Enable/disable output of Access Address bit sequence via data_out port depending on aa_output_sel according to the following table:
<table>
<tr><th>AA_PLAYBACK</th><th>AA_OUTPUT_SEL</th><th>ACTIONUNITS</th></tr>
<tr><td>0</td><td>X</td><td>Only PDU bits output via data_out port. No AA bits</td></tr>
<tr><td>1</td><td>0</td><td>Demodulated AA bits followed by PDU bits are output via data_out port.</td></tr>
<tr><td>1</td><td>1</td><td>Matched AA bits followed by PDU bits are output via data_out port.</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

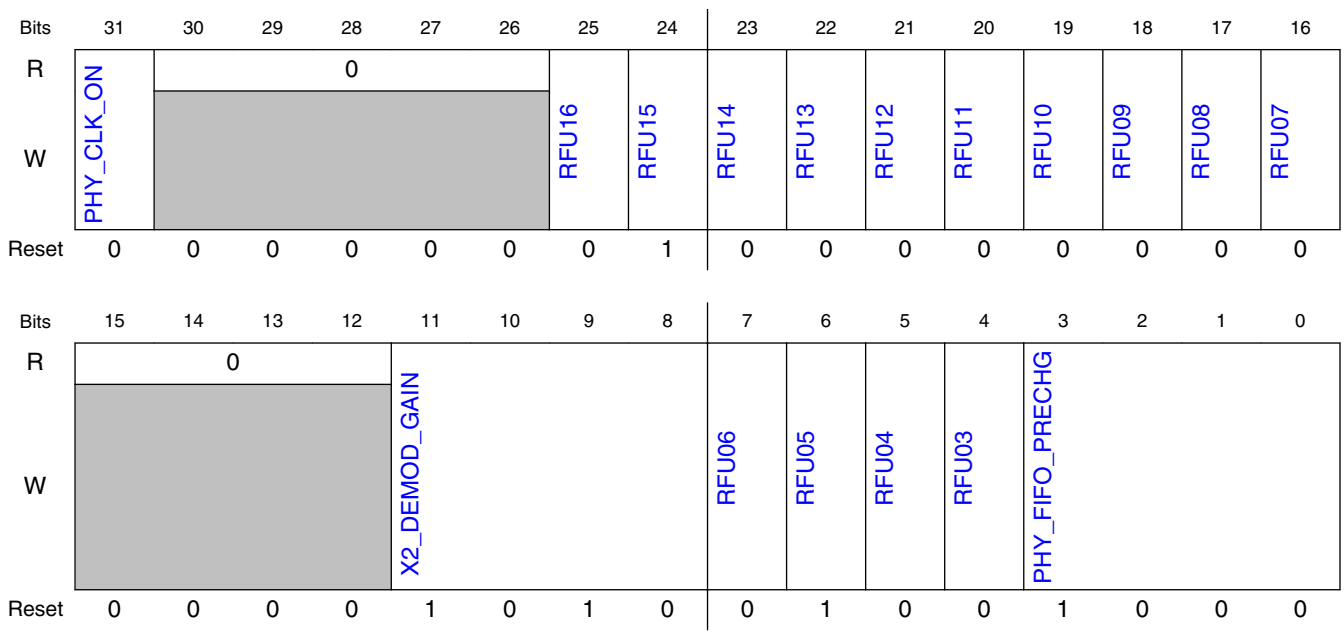
Field	Function
	Note: This bit must be set to 1 for Generic FSK, and must be set to 0 for BLE.
0 —	Reserved

45.3.9.2.6 PHY CONFIGURATION REGISTER 2 (CFG2)

45.3.9.2.6.1 Offset

Register	Offset
CFG2	24h

45.3.9.2.6.2 Diagram



45.3.9.2.6.3 Fields

Field	Function
31 PHY_CLK_ON	Force PHY Clock On (testmode) 0b - PHY clock is enabled by TSM output: rx_phy_en 1b - PHY clock is forced on at all times
30-26 —	Reserved

Table continues on the next page...

Field	Function
25 RFU16	Reserved for future use.
24 RFU15	Reserved for future use.
23 RFU14	Reserved for future use.
22 RFU13	Reserved for future use.
21 RFU12	Reserved for future use.
20 RFU11	Reserved for future use.
19 RFU10	Reserved for future use.
18 RFU09	Reserved for future use.
17 RFU08	Reserved for future use.
16 RFU07	Reserved for future use.
15-12 —	Reserved
11-8 X2_DEMOD_GAIN	X2_DEMOD_GAIN Gain parameter used in the symbol demodulator. The unsigned fixed-point gain is 4 bits wide representing a range of representable gain values: [0, ..., 1-1/2^4] = {0000, ..., 1111}. X2_DEMOD_GAIN is a function of the PHY modulation, scheme, data rate, and the receiver chain filtering characteristics and is chosen to minimize the demodulation bit error rate. The table below shows the recommended values for an assortment of FSK schemes. They are determined from Simulink model simulations. For more details, please refer to the Multi-PHY chapter in the 2.4GHz Radio Gen 2 ADD.

Modulation Scheme	Data Rate (kbps)	X2_DEMOD_GAIN value	X2_DEMOD_GAIN bit representation
BLE (GFSK BT=0.5, h=0.5)	1000	0.6250	1010
ANT (GFSK BT=0.5, h=0.32)	1000	0.1875	0011
GFSK BT=0.5, h=0.7	500	0.6250	1010
GFSK BT=0.3, h=0.5	1000	0.6250	1010
MSK	500	0.6250	1010
GFSK BT=0.5, h=0.5	250	0.2500	0100
GFSK BT=0.5, h=1.0	250	0.5000	1000

Table continues on the next page...

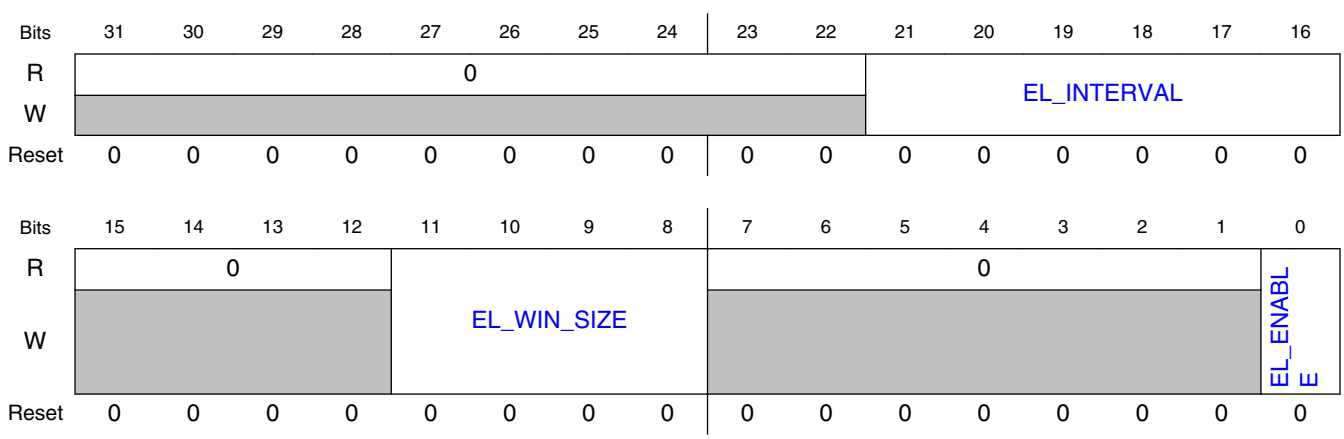
Field	Function			
	Modulation Scheme	Data Rate (kbps)	X2_DEMOD_GAIN value	X2_DEMOD_GAIN bit representation
	GFSK BT=0.7, h=0.5	1000	0.5000	1000
7 RFU06	Reserved for future use.			
6 RFU05	Reserved for future use.			
5 RFU04	Reserved for future use.			
4 RFU03	Reserved for future use.			
3-0 PHY_FIFO_PRECHG	PHY FIFO Precharge Level Indicates the precharge depth of the output FIFO.			

45.3.9.2.7 PHY EARLY/LATE CONFIGURATION REGISTER (EL_CFG)

45.3.9.2.7.1 Offset

Register	Offset
EL_CFG	28h

45.3.9.2.7.2 Diagram



45.3.9.2.7.3 Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31-22
—</td><td>Reserved</td></tr>
<tr><td>21-16
EL_INTERVAL</td><td>EL_INTERVAL
No. of FSK/IEEE 802.15.4 symbols between successive EL operation windows. Valid in both FSK and IEEE 802.15.4 modes.
<table><tr><th>Parameter</th><th>BLE</th><th>802.15.4</th></tr><tr><td>EL_WIN_SIZE[3:0]</td><td>8</td><td>3</td></tr><tr><td>EL_INTERVAL[5:0]</td><td>32</td><td>7</td></tr></table></td></tr>
<tr><td>15-12
—</td><td>Reserved</td></tr>
<tr><td>11-8
EL_WIN_SIZE</td><td>EL_WIN_SIZE
Number of successive FSK/IEEE 802.15.4 symbols over which one EL operation occurs. Valid in both FSK and IEEE 802.15.4 modes.</td></tr>
<tr><td>7-1
—</td><td>Reserved</td></tr>
<tr><td>0
EL_ENABLE</td><td>EL_ENABLE
Enable/disable EL mechanism during PDU/PSDU demodulation
0b - Disable Early/Late
1b - Enable Early/Late</td></tr>
</table>

45.3.9.2.8 PHY NETWORK ADDRESS FOR BSM (NTW_ADR_BSM)

45.3.9.2.8.1 Offset

Register	Offset
NTW_ADR_BSM	2Ch

45.3.9.2.8.2 Diagram

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

45.3.9.2.8.3 Fields

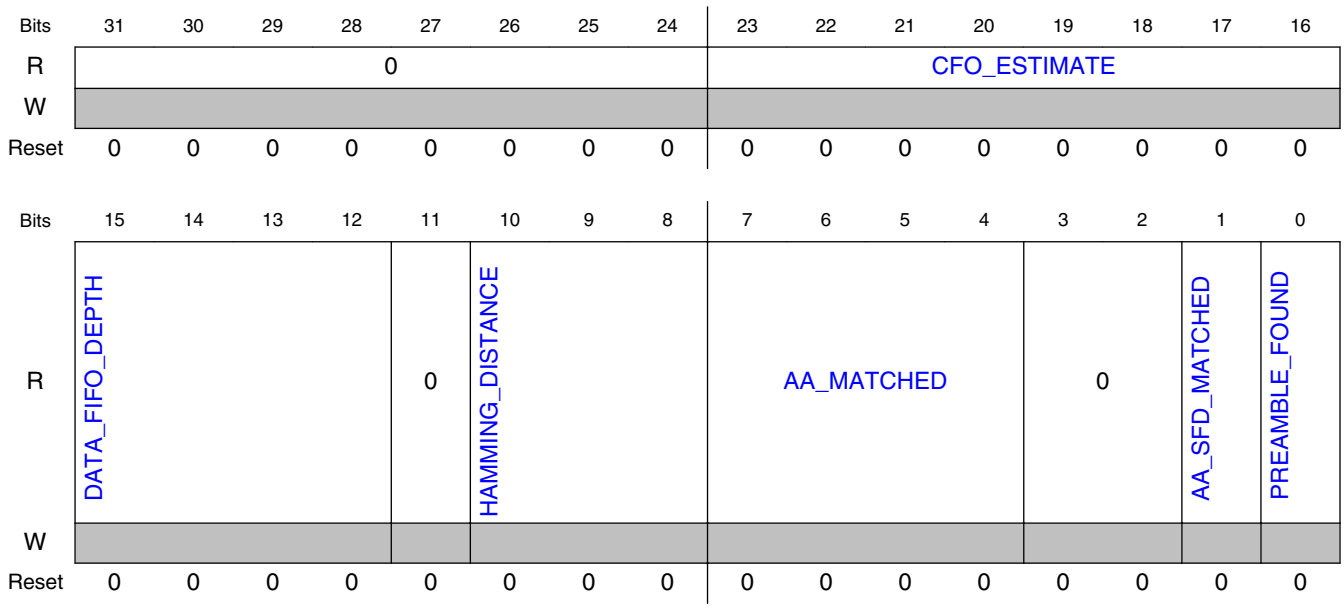
Field	Function
31-0 NTW_ADR_BSM	NTW_ADR_BSM PHY will search for this 32-bit Access Address when PHY_CFG1[BSM_EN_BLE]=1

45.3.9.2.9 PHY STATUS REGISTER (STATUS)

45.3.9.2.9.1 Offset

Register	Offset
STATUS	30h

45.3.9.2.9.2 Diagram



45.3.9.2.9.3 Fields

Field	Function
31-24 —	Reserved
23-16 CFO_ESTIMATE	Carrier Frequency Offset Estimate Most recent estimate for Carrier Frequency Offset. The multiplication factor is 7812: i.e., CFO_ESTIMATE x 7812 = actually frequency offset in Hz (ideally).
15-12 DATA_FIFO_DEPTH	DATA FIFO DEPTH Instantaneous depth of the PHY output FIFO. The difference between the FIFO write pointer and read pointer.
11 —	Reserved
10-8 HAMMING_DISTANCE	HAMMING DISTANCE Valid only in FSK mode. Indicates hamming distance between observed AA pattern and the candidate AA pattern that was found to be the best match.
7-4 AA_MATCHED	Access Address Matched All bits reset in IDLE state. When any bit asserted, indicates which of the 4 Network Addresses has been matched. found. Valid only in FSK mode. 0000b - No Network Address has matched 0001b - Network Address 0 has matched 0010b - Network Address 1 has matched 0100b - Network Address 2 has matched 1000b - Network Address 3 has matched
3-2 —	Reserved

Table continues on the next page...

Field	Function
1 AA_SFD_MATCHED	Access Address or SFD Found Reset in IDLE state or when activate_search is de-asserted. Asserted when the AA/SFD is found.
0 PREAMBLE_FOUND	Preamble Found Reset in IDLE state or when *activate_search* is de-asserted. Asserted when the preamble is found and the coarse symbol timing is determined. If the subsequent AA/SFD search fails and the receiver resumes CTS, this signal will be reset.

45.4 Transceiver Analog

45.4.1 Introduction

The 2.4 GHz radio is comprised of both an analog and digital portion. The analog portion of the radio provides the local oscillator (LO), transmit modulation, and receiver down conversion, along with the various circuits needed to support those functions.

The analog block also provides a clock to the digital domain that is to be used as a reference. This clock is derived from an external reference and, in the case of a crystal, should be trimmed to be as accurate as possible. The two supported frequencies for this reference are 26 MHz and 32 MHz.

RF input and output are combined with an integrated balun located on the device. Thus, only a single RF port is needed.

It is also of note that the analog contains several LDO circuits to supply various blocks within the design.

The analog radio is configured and commanded by the digital domain using both static configuration signals and dynamic control signals.

45.4.1.1 Features

The 2.4 GHz analog radio block includes the following features:

- Fractional-N Frequency Synthesizer PLL shared by receive and transmit
- Integrated balun
- Transmitter supporting constant envelope modulation for 2.4 GHz ISM and 2.36 GHz MBAN frequency bands (2360 to 2483.5 MHz)

- Direct PLL modulation for transmit
- Direct conversion reciever for 2.4 GHz ISM and 2.36 GHz MBAN frequency bands (2360 to 2483.5 MHz)
- Receive line-up comrised of a Low Noise Amplifier & Mixer, Baseband amplifier and filter, and sigma-delta ADC
- Direct analog diagnostic lines for debug and test
- Multiple LDOs to power analog circuits from line supply
- XO (REF_OSC) ciruit to use and external crystal and provide a reference clock internally at 26 MHz or 32 MHz
- AuxPLL to provide a 2x reference clock
- DC correction DACs for the TZA and BBA stages
- Programmable receiver gains in both the LNA and BBA stages

45.4.1.2 Block diagram

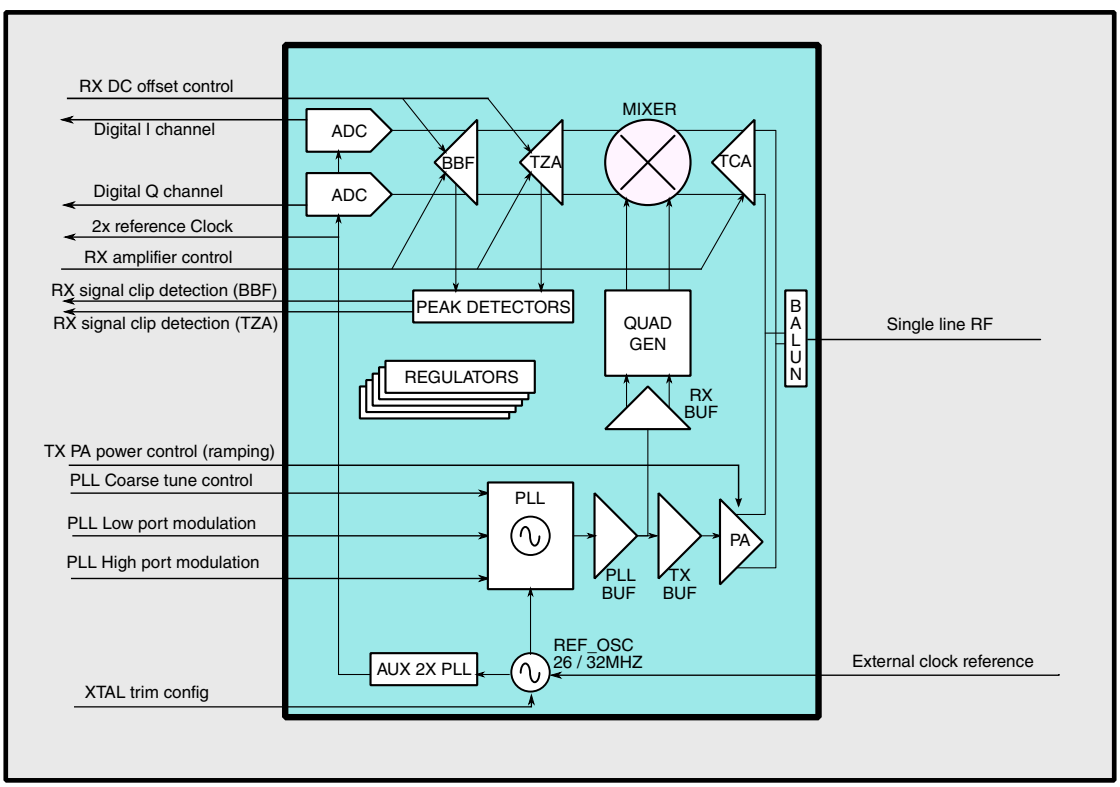


Figure 45-99. Analog Radio Block diagram

45.4.2 Functional description

The following section briefly describes the functionality of the 2p4GHz analog radio. It is not intended to be a complete description as many of the details are covered by descriptions of the control signals that are provided from the digital domain.

The analog block serves several purposes. Primarily, it provides the necessary circuitry to transmit at 2.4 GHz and receive in the same bands. When configured for transmission, the PLL is locked to the desired frequency (band and channel) as controlled by the PLL digital module. The digital radio then modulates the high port and low port of the PLL to generate the desired RF transmission. For receiver operation, the PLL is locked to the desired band and channel just as in transmit. The mixer provides a down-converted signal at baseband. The signal then goes through the baseband amplifier and is presented to the ADC. The ADC then converts the analog signal to a digital format, where it is observed by the digital domain. The receiver uses a 2x frequency clock to convert, so this clock is also provided to the digital domain from the AuxPLL. The receive path also contains several configurable parameters. For the gain, the LNA and BBA stages can be adjusted by signals from the digital domain. This is typically performed by the AGC circuit. DC offset can be corrected by controlling offset DACs at both stages, as well.

Both transmit and receive utilize an integrated balun. This results in a single RF port for the device and eliminates the need for a board level balun.

Several LDO blocks within the analog block provide the necessary supply to corresponding circuits. There is an LDO for the BBA, etc.

The XO block uses a reference crystal (typically) to produce a stable reference clock for the analog blocks and the digital domain. More details are provided in the section describing this block.

To assist in measurements and debug, multiple diagnostic analog signals are provided. These signals can be configured by inputs from the digital domain and then observed externally or with an ADC resident in the device.

45.4.2.1 XO Block

The XO block generates a 26 MHz or a 32 MHz reference clock for the digital domain, PLL, DAC, and AuxPLL blocks of the radio. It generates the output clocks from a crystal mounted externally to the device.

This block is alternatively known as REF_OSC, OSC, or XTAL, ocassionally.

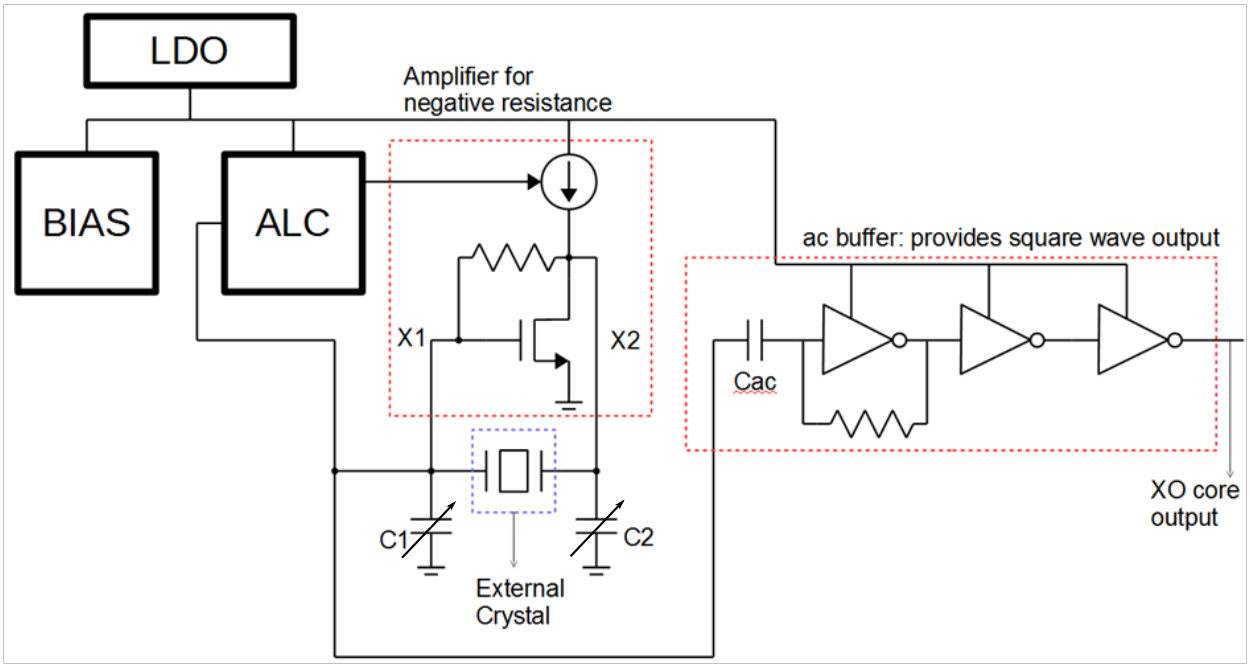


Figure 45-100. XO circuit diagram

45.4.2.1.1 XTAL Trim

To enable optimum performance, it is required to trim the XO such that an accurate clock is produced based on the external crystal used. C1 and C2 are tuneable and intended for this purpose. They can be adjusted by configuring the XTAL_TRIM bits in the digital domain.

45.5 Link Layer

45.5.1 BLE Link Layer

45.5.1.1 BLE Register Descriptions

BLE base address: 4005_B000h

45.5.1.1.1 Platform Registers

Platform Register Descriptions for the Bluetooth Link Layer

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	15:11		XX
conn_index[2:0]	10:8	Connection index to specify the command is for which connection engine.	0
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 appendix 1 for details of the registers to be set before setting these instructions.	00

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	0x0000

				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.	

Field	Bit	Description	Reset
Reserved	15:9	Not used.	XX
Dtm_intr_clr	8	Clear DTM packet reception interrupt. Write to the register with this bit set to 1, clears the interrupt source.	
Reserved	7:6	Not used.	
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
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	15:10	Not used.	XX
TSM_intr	9	TSM interrupt when BLE protocol is active.	0
Dtm_intr	8	DTM RX packet reception interrupt.	0

Table continues on the next page...

Reserved	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 the INIT_STATUS register. This bit is cleared, when firmware clears ALL interrupts by writing to the INIT_INTR_CLEAR register.	0
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	0

		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.			

Event enable register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 enable the interrupts back. This ensures no new interrupts is missed while firmware is processing one.	0x0000

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.	0

Table continues on the next page...

		1 – enable sleep mode exit event to interrupt the firmware. 0 – disable sleep mode exit interrupt to firmware.	
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
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	0

Table continues on the next page...

		field, which is used every time hardware does an auto-wakeup before the next wakeup instant.	
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
Retain_in_dsm2	8	“1” indicates Connection RAM to be retained during DSM2.	0
Reserved	7:5	Reserved for future use.	0
Osc_startup_delay[4:0]	4:0	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 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[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)	00

Wakeup Configuration2 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x278	0x4F0	WAKEUP_CONFIG2	RW	Wakeup configuration to configure the various offsets while waking up from sleep mode or deep sleep mode.	0x00

Field	Bit	Description	Reset
Reserved	15:7	Not used.	XX
Osc_startup_delay[11:5]	6:0	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 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[11:5] is number of slots(625us)`	00

Note: In case of firmware 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 *_NEXT_INSTANT registers. This value is used only when	0

		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.	0x0080

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
sm_intr_en	12	Enable SM exit interrupt. 1 – enable, 0 – disable.	0

Table continues on the next page...

Field	Bit	Description	Reset
		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.	
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
LLH_idle (READ ONLY field)	7	Indicates if hardware is doing any transmit/receive operation or there is a pending interrupt	1

Field	Bit	Description	Reset
		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).	
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...

		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 the operation of the hardware.	0x0000

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
RFU	15:8	Not used	0x0
bb_clk_freq_minus_1	7:3	The frequency information (FREQ – 1) of the clock input to LL Hardware is configured in this register by the firmware during initialization. This information is used inside LL Hardware to derive timing information for the Bluetooth operations. For example, if the frequency of the input clock is 12 MHz this field shall be programmed to 0xB. (1 less than frequency)	0x0
RFU	2:0	Not used	0x0

DSM configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X51A	0XA34	DSM_CONFIG	RW	DSM configuration register.	0x0008

Field	Bit	Description	Reset
wl_clk_gate_en	15	Whitelist block clock gate enable. 1 = Enable, 0 = Disable. Enables gating of clock to the whitelist 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.	0x0
enable_conn_clk_ctrl[7:0]	[14:7]	Each bit corresponds to one connection engine (bit 7 for conn engine 0). Each of the bits if set to 1 will enable LLH to turn on corresponding CONNECTION engine clock on EXIT_SM command from firmware. To use this feature CON-NECTION NAP modes shall be enabled in LLH.	0x0
enable_init_clk_ctrl	6	If this bit is set to '1' then LLH will turn on INIT engine clock on EXIT_SM command from firmware. To use this feature INIT NAP modes shall be enabled in LLH.	0x0
enable_scan_clk_ctrl	5	If this bit is set to '1' then LLH will turn on SCAN engine clock on EXIT_SM command from firmware. To use this feature SACN NAP mode shall be enabled in LLH.	0x0
enable_adv_clk_ctrl	4	If this bit is set to '1' then LLH will turn on ADV engine clock on EXIT_SM command from firmware. To use this feature ADV NAP mode shall be enabled in LLH.	0x0
enable_eng_clk_ctrl	3	Configuration bit for turning off all engine specific clocks after wakeup from SHUTDOWN (after register restore is complete). If this bit is set to '1' then LLH will turn off all engine specific clocks after restore_done. To use this feature all NAP modes shall be enabled in LLH.	0x1
restore_done_bypass	2	Configuration bit for bypassing wait for restore_done signal to exit from RESTORE state. If this bit is set to '1' then power control FSM will move to next	0x0

Table continues on the next page...

		state without waiting for restore_done from RESTORE state and if this bit is set to '0' then power state machine will wait in RESTORE state till restore_done comes. This bit can be effectively used to reduce wakeup time if restore gets completed in one LP clock cycle.	
store_done_bypass	1	Configuration bit for bypassing wait for store_done signal to enter in to SHUTDOWN. If this bit is set to '1' then power control FSM will move to next state without waiting for store_done from STORE state and if this bit is set to '0' then power state machine will wait in STORE state till store_done comes. This bit can be effectively used to reduce SHUTDOWN entry time if store gets completed in one LP clock cycle.	0x0
dsm_config	0	Configuration bit for DSM SHUTDOWN LLH store and restore control. If this bit is set to '1' then LLH will do the store and restore of its registers on its own during entry and after exit from shutdown. To use this feature DSM SHUTDOWN mode is to be supported and also LLH shall have access to RETENTION RAM.	0x0

Note To enable dynamic power saving though selective clock gating of different engines LL firmware will set the configuration bits corresponding to BT procedures which are active. This will ensure that clocks to only active blocks are enabled during EXIT_SM command from LL firmware to LL hardware. User can over-ride this mode by setting the bits.

Nearest Instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0X520	0XA40	NEAREST_INST	RO	This register reports the next immediate instant in terms of bt_clock value (actual value being reported is INSTANT – 3) where hardware has any scheduled BT activity. Firmware reads this register along with the TIM_COUNTER_L register to calculate the possibility of power save mode entry.	0x0000

Field	Bit	Description	Reset
global_nearest_instant	15:0	This field holds the immediate future value of BT_CLOCK where a scheduled Bluetooth activity is supposed to happen. When no Bluetooth activity is running the register will hold current BT_CLOCK value – 3.	0x0000

Auto DSM Configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X274	0X4E8	AUTO_DSM_CONFIG	RW	Auto DSM configuration register.	0x0000

Field	Bit	Description	Reset
Not Used	15:6	Not Used	0x0
Auto_entry_fsm_store_val	5:4	Store/Restore of Auto entry FSM states.	0x0
Dms3_mode_en	3	Enable DSM3 mode for Auto DSM entry feature. 1 – enable, 0 – disable.	0x0
Dsm2_mode_en	2	Enable DSM2 mode for Auto DSM entry feature. 1 – enable, 0 – disable.	0x0
Fw_wakeup_en	1	Enable wakeup at every FW_WAKEUP_INSTANT by firmware when HW is in sleep mode. 1 – enable, 0 – disable.	0x0
Dsm_auto_sleep_en	0	Reserved	0x0

Firmware wakeup Instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X276	0X4EC	FW_WAKEUP_INSTANT	RW	Wakeup instant register. Program the 16-bit reference clock value for auto-wakeup from deep sleep mode.	0x0000

Field	Bit	Description	Reset
Fw_wakeup_instant	15:0	This field holds the clock instant value when LLH need to exit auto dsm mode.	0x0000

45.5.1.1.2 Advertising Channel Registers

Advertising Channel Register Descriptions for the Bluetooth Link Layer

##*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 register before issuing start advertise command. The fields in this register correspond to the fields in the LE_Set_Advertising_Parameters HCI command.	0x00E0

Field	Bit	Description	Reset
##peer_rx_txaddr (Read Only)	15	Transmit address field of the received packet indicating the peer address type as public/ random. This field is used by firmware to report peer_addr_type parameter in the connection complete event.	0

Table continues on the next page...

		0 – addr type is public. 1 – addr type is random.	
##peer_addr_resolved_reg (Read Only)	14	Address resolution status of the received packet if the received address was RPA. 0 – RPA resolution fail 1 – RPA resolution pass. Else this bit is set to zero.	0
#slv_index	13:11	Slave index is programmed by FW before initiation procedure for selecting connection engine in slave role connection. Number of supported engines is 8. (Index from 0x0 to 0x7)	0
*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
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.	7

		Bit 6 - enable channel 38. Bit 5 - enable channel 37.	

Field	Bit	Description	Reset
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
Reserved	0	Reserved for future use.	0

Advertising Interval register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

Table continues on the next page...

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
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
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

Table continues on the next page...

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 register before issuing start scan command.	0x0010

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.	0X0010

		Range: 0x0004 to 0x4000 Default: 0x0010 (10 ms) Time = N * 0.625 msec Time Range: 2.5 msec to 10.24 sec .	

##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	15:7	Not used.	XX
scan_tx_addr[1]	6	MSB of the device's Own address. This bit along with scan_tx_addr[0] decides the own address used for scanner.	0
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. A connectable Directed advertising packet not containing the scanner's device address is ignored.	0

Table continues on the next page...

		0x01 - Accept advertising packets from only devices in the whitelist . A connectable Directed advertising packet not containing the scanner's device address is ignored. 0x10 - Accept advertising packets from any device. A connectable Directed advertising packet is not ignored if the InitA is a resolvable private address. 0x11 - Accept advertising packets from only devices in the whitelist. A connectable Directed advertising packet is not ignored if the InitA is a resolvable private address. Adv_direct_ind packets which are not addressed to this device are ignored.	
scan_type	2:1	0x00- passive scanning. (default) 0x01- active scanning. 0x10- RFU 0x11- RFU	0
scan_tx_addr[0]	0	Device's own address type. 1 - addr type is random. 0 - addr type is public. Note : Scan_tx_addr[1:0] 0x0 – use public address 0x1 – use random address 0x2 – use RPA if matching entry is available in Resolving list else use public address. 0x3 - use RPA if matching entry is available in Resolving list else use random address.	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	0x0000

				interrupts set in the SCAN_STATUS register. One or more interrupts can be cleared in a single write operation.	

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.	

Table continues on the next page...

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	0X0000

		Time Range: 2.5 msec to 10.24 sec .	

##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	15:10	Not used.	XX
peer_rpa_resolved_capt (Read only)	9	If received peer address is an RPA, this bit represents the RPA resolution status. ie., 1 – RPA resolution pass. 0 – RPA resolution fail. Else this bit is set to zero.	0
init_tx_addr[1]	8	MSB of the device's Own address type. This bit along with init_tx_addr[0] decides the own address of the device.	0
#Init_conn_index	7:5	Firmware programs the index of the conn engine for the master connection at init start. Index can be in the range 0x0 to 0x7 depending on the number of connections supported.	0x0
Reserved	4	Not used	x
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	0

Table continues on the next page...

		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.	
init_rx_addr[1]	2	MSB of the device's peer address type (FW programmed). This bit along with init_rx_addr[0] decides the peer address of the device.	0
rx_addr/rx_tx_addr init_rx_addr[0]	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. LSB of the device's peer address type (FW programmed). Init_rx_addr[1:0] – 0x0 - public address 0x1 - random address 0x2 - public identity address 0x3 - random(static) identity address	0

Tx_addr init_tx_addr[0]	0	Own address type. 1- addr type is random. 0-addr type is public. LSB of the device's Own address type. Init_tx_addr[1:0] - 0x0 – use public address (addr type is zero) 0x1 – use public address (addr type is one)	0

		0x2 – use RPA if matching entry is available in Resolving list else use public address. 0x3 - use RPA if matching entry is available in Resolving list else use random address.	

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 be cleared in a single write operation.	0x0000

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x28	0x050	INIT_STATUS	RO	Shows the status of the interrupt. This register is read	0x0000

				by firmware to learn the pending initiator interrupts that are set and are to be served.	

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
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

#Initiator Anchor Point register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x100	0x200	INIT_ANCHOR_POINT	RW	Firmware programs this register (while programming INIT parameters) with the	0x0000

				instant (value of bt_clock) at which the initiator should start the procedure for the first time. This will give firmware a better control in effectively scheduling the new connections in the multiple connection scenarios.	

Field	Bit	Description	Reset
Init_anchor_point	15:0	The value of bt_clock (625us period) at which initiator should start the procedure.	0

Device Random address lower register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x2C	0x058	DEV_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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x2E	0x05C	DEV_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

Table continues on the next page...

16-bit	32-bit				
0x30	0x060	DEV_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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

#Initiator Peer Address Lower Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x66	0x0CC	INIT_PEER_ADDR_L	RW	Lower 16 bit of init peer address. This register is used only if multiple connections are supported.	0x0000

Field	Bit	Description	Reset
Init_peer_addr	15:0	Lower 16 bit of the 48 bit init peer address.	0

#Initiator Peer Address Middle Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x68	0x0D0	INIT_PEER_ADDR_M	RW	Middle 16 bit of init peer address. This register is used only if multiple connections are supported.	0x0000

Field	Bit	Description	Reset
Init_peer_addr	15:0	Middle 16 bit of the 48 bit init peer address.	0

#Initiator Peer Address Upper Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x6A	0x0D4	INIT_PEER_ADDR_H	RW	Lower 16 bit of init peer address. This register is used only if multiple connections are supported.	0x0000

Field	Bit	Description	Reset
Init_peer_addr	15:0	Upper 16 bit of the 48 bit init peer address.	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.

INIT_PEER_ADDR is used only for multiple connection support where peer address for initiation is kept at INIT_PEER_ADDR register and ADV peer address is kept/read from PEER_ADDR register.

White List address type register lower word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xA2	0x144	WL_ADDR_TYPE_L	RW	Stores the address type of the device addresses stored in whitelist	0x0000

Field	Bit	Description	Reset
type	15:0	16 address type bits corresponding to first 16 device address stored in whitelist.	0

		1-Address type is random. 0-Address type is public.	

White List address type register upper word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xA4	0x148	WL_ADDR_TYPE_U	RW	Stores the address type of the device addresses stored in whitelist	0x0000

Field	Bit	Description	Reset
	15:10	Reserved for future use	
type	9:0	10 address type bits corresponding to 17 to 26 device address stored in whitelist. 1-Address type is random. 0-Address type is public.	0

White list enable register lower word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xA6	0x14C	WL_ENABLE_L	RW	Stores the valid entry bit corresponding to each of the device address stored in the whitelist.	0x0000

Field	Bit	Description	Reset
en	15:0	Stores the valid entry bit corresponding to each of the first 16 device addresses stored in the whitelist memory in the hardware. 1-White list entry is valid. 0-White list entry invalid.	0

White list enable register upper word

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0xA8	0x150	WL_ENABLE_U	RW	Stores the valid entry bit corresponding to each of the device address stored in the whitelist.	0x0000

Field	Bit	Description	Reset
	15:10	Reserved for future use	
en	9:0	Stores the valid entry bit corresponding to each of the 17 to 26 device addresses stored in the whitelist memory in the hardware. 1-White list entry is valid. 0-White list entry invalid.	0

Advertising data transmit FIFO

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 address 16'h74	Data byte 1	Length of scan response host data
	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 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.	0x0000

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

Table continues on the next page...

16-bit	32-bit				
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 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.	0x0006

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 channel configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x27E	n/a	ADVCH_CONFIG	RW	Advertiser channel procedure configuration register.	0x0

Field	Bit	Description	Reset
Reserved	15:1	Not used.	0
Bypass_arb	0	Firmware set this bit when only one procedure (either Advertising, Scanning, Initiation or Connection) is running to indicate hardware that arbitration is not required. 0: Arbitration is required 1: Bypass arbitration	0

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	0xE07F

Field	Bit	Description	Reset
scan_channel_map	15:13	Advertising channels that are enabled for scanning operation.	111

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	Not used.	X
backoff_enable	11	Enable random backoff feature in scanner. 1-enable. 0-disable.	0
Reserved	10:5	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. 15- Enables channel 39 for use. 14- Enables channel 38 for use. 13- Enables channel 37 for use.	0
Reserved	12:8	Not used.	XX

Table continues on the next page...

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
0X462	0x8C4	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], 2nd write - [31:16], 3rd write - [47:32] bits of the device address.	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

Whitelist end address register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X4FE	0x9FC	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 26th 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
0X700	0XE00	CONN_TXMEM_BASE_ADDR	WO	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	0x0000

				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.	

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X400	0X800	CONN_RXMEM_BASE_ADDR	RO	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 read from Rx memory are read 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,	0x0000

				before initiating connection. After slave connection this value can be obtained by reading this register on the advertiser side.	

Field	Bit	Description	Reset
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 values	0x0000

				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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1E8	0x3D0	CONN_REQ_WORD4	RW	This field defines the 16 bits of the transmit window offset	0x0000

				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.	

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	0x0000

				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.	

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
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. '1' indicates the corresponding data channel is used and '0' indicates the channel is unused.	Data_channels

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

##RPA timer interval register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x582	0xB04	RPA_TIMER_INTERVAL	RW	The RPA timer interval register stores the RPA refresh time interval. (FW programmed).	0x0000

Field	Bit	Description	Reset
rpa_timer_interval_val	15:0	This field holds the value of the RPA refresh time interval in seconds.	0

##Privacy Configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x108	0x210	LL_PRIVACY_CONFIG	RW	This register fields can be set to enable/disable Privacy feature in LLH.	0x0000

Field	Bit	Description	Reset
Reserved	15:2	Reserved for future use.	0
allow_rpa_in_whitelist	1	Allow initiator to receive advertising packets even if peer address resolution fails by adding the received advertiser RPA to the whitelist and to the resolving list as per Errata 6984.	0
privacy_config	0	Whenever this bit is set LLH will try to resolve the address for all ADVCH procedures.	0

##RPA selection index register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x10A	0x214	CURRENT_LOCAL_IRK_INDEX	RW	This register is programmed by firmware before advertising and Initiation procedures. This	0xFF

				bit is used to indicate the local IRK to be used by HW during advertising and Initiation procedures.	

Field	Bit	Description	Reset
Reserved	15:8	Reserved for future use.	0
init_local_irk_index	7:4	This field holds the resolving list index corresponding to the IRK to be used for the INIT procedure. This field is invalid when white list is enabled. Value 0xF – valid IRK for INIT procedure is not set in resolving list.	0xF
adv_local_irk_index	3:0	This field holds the resolving list index corresponding to the IRK to be used for the ADV procedure. This field is also valid when white list is enabled. Value 0xF – valid IRK for ADV procedure is not set in resolving list.	0xF

##Peer identity address type register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x10C	0x218	PEER_IDENTITY_ADDR_TYPE	RW	This register stores the peer identity address type (Public or Random) corresponding to the IRK pair populated in the resolving list. The value is valid if the resolving list entry is valid.	0x0

Field	Bit	Description	Reset
Reserved	15:4	Reserved for future use.	0
peer_id_addr_type	3:0	Peer_id_addr_type [i] = 0/1 => ith resolving list entry corresponds to a peer whose address is public/random.	0

##Resolving list entry valid register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x10E	0x21C	RES_LIST_VALID	RW	This register is used to indicate to HW which resolving list entries are valid/ invalid.	0x0

Field	Bit	Description	Reset
Reserved	15:4	Reserved for future use.	0
device_valid_entry	3:0	Device_valid_entry [i] = 0/1 => ith resolving list entry is invalid/valid.	0

##Advertising parameters 2 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x0A	0x14	ADV_PARAMS_2	RW	This register stores the own address type for the Advertising procedure.	0x0

Field	Bit	Description	Reset
Reserved	15:2	Reserved for future use.	0
adv_tx_addr	1:0	Device Own_address_type field. 0x0 - Own address type is public address. 0x1 - Own address type is random address. 0x2 - Own address type is RPA. If resolving list contains no matching entry, use public address. 0x3 - Own address type is RPA. If resolving list contains no matching entry, use random address.	0

##Local IRK n registers

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x1100-0x11FE	-	L_IRK_n_x	RW	These registers store the local IRK value for nth resolving list entry. IRK value is of 128 bits and stored in Little Endian format (lowest address register has the LSW).	0x00000000000000000000000000000000

Field	Bit	Description	Reset
dev_localirk[n]	127:0	This field has the 128 bit value of local IRK for nth resolving list entry. Note: Refer to appendix 6 for full addressing with respect to each resolving entry.	0

##Peer IRK n registers

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1200-0x12FE	-	P_IRK_n_x	RW	These registers store the peer IRK value for nth resolving list entry. IRK value is of 128 bits and stored in Little Endian format (lowest address register has the LSW).	0x00000000000000000000000000000000

Field	Bit	Description	Reset
dev_peerirk[n]	127:0	This field has the 128 bit value of peer IRK for nth resolving list entry. Note: Refer to appendix 6 for full addressing with respect to each resolving entry.	0

##Peer identity address n registers

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1300-0x135E	-	PEER_ID_ADDR_n_x	RW	These registers store the peer identity address for nth resolving list entry. Identity address is of 48 bits and stored in Little Endian format (lowest address register has the LSW).	0x000000000000

Field	Bit	Description	Reset
dev_identity_addr[n]	47:0	This field has the 48 bit value of peer ID address for nth resolving list entry. Note: Refer to appendix 6 for full addressing with respect to each resolving entry.	0

##Local RPA n registers

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1360-0x13BE	-	L_RPA_n_x	RO	These registers store the local RPA generated by the HW using the local IRK of the nth resolving list entry. RPA is of 48 bits and stored in Little Endian format (lowest address register has the LSW).	0x0000000000

Field	Bit	Description	Reset
self_rpa_list_n	47:0	This field has the 48 bit value of local RPA for nth resolving list entry. Note: Refer to appendix 6 for full addressing with respect to each resolving entry.	0

##Peer RPA n registers

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x13C0-0x141E	0x1CC0 – 0x1CC8	P_RPA_n_x	RO	These registers store the peer RPA generated by the HW using the peer IRK of the nth resolving list entry. RPA is of 48 bits and stored in Little Endian format (lowest address register has the LSW).	0x0000000000

Field	Bit	Description	Reset

Table continues on the next page...

peer_rpa_list_n	47:0	This field has the 48 bit value of peer RPA for nth resolving list entry. Note: Refer to appendix 6 for full addressing with respect to each resolving entry.	0

##RPA timer wrap count register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x586	0xB0C	RPA_TIMER_WRAP_COUNT	RO	The RPA timer wrap count register stores the wrap count value. A wrap is a complete bt_clock roll from 0x0000 to 0xFFFF.	0x0

Field	Bit	Description	Reset
Reserved	**15:10**		**0**
rpa_timer_wrap_count	9:0	This field holds the bt_clock wrap count value.	0

##RPA timer current wrap register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x584	0xB08	RPA_TIMER_CURRENT_WRAP	RO	The RPA timer current wrap register stores the current wrap count value.	0x0

Field	Bit	Description	Reset
Reserved	15:12		0
rpa_timer_current_wrap	11:2	This field holds RPA timer's current wrap count value.	0
rpa_timer_en	1	Indicates whether RPA timer is ON in hardware. 1 – on 0 – off	0
wrap_valid	0	Indicates whether RPA timer's warp is valid. 1 – wrap is valid 0 – wrap is invalid	0

##RPA timer next instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x588	0xB10	RPA_TIMER_NEXT_INSTANT	RO	Holds the next RPA timer instant value.	0x0

Field	Bit	Description	Reset
Reserved	**15:10**		**0**
rpa_timer_next_instant	9:0	Holds the next RPA timer instant value. This instant is valid when wrap valid = 1.	0

##RPA LFSR seed lower register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x9C	0x138	RPA_LFSR_L	RW	Lower seed value of the LFSR for RPA generation.	0x0

Field	Bit	Description	Reset
Rpa_lsfr_l	15:0	Firmware can writes to this register to update the seed value of the LFSR for random RPA generation after the reset.	0

##RPA LFSR seed middle register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x9E	0x13C	RPA_LFSR_M	RW	Middle seed value of the LFSR for RPA generation.	0x0

Field	Bit	Description	Reset
Rpa_lsfr_m	15:0	Firmware can writes to this register to update the seed value of the LFSR for random RPA generation after the reset.	0

##RPA LFSR seed higher register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xA0	0x140	RPA_LFSR_H	RW	Upper seed value of the LFSR for RPA generation.	0x0

Field	Bit	Description	Reset
Reserved	15:12	RFU	**0**
Rpa_lsfr_u	11:0	Firmware can writes to this register to update the seed value of the LFSR for random RPA generation after the reset.	0

##Privacy Mode register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x280 – 0x28E	0x500 – 0x51C	PRIVACY_MODE_0 – PRIVACY_MODE_7	RW	Mode of the privacy address added for each entry in the resolving list.	0x0

Field	Bit	Description	Reset
Reserved	15:1	RFU	**0**
Privacy_mode	0	This bit defines the mode of the device is network privacy or device privacy. 0 – Network Privacy Mode 1 – Device Privacy Mode	0

45.5.1.1.3 Data Channel Registers

Data Channel Register Descriptions for the Bluetooth Link Layer

Transmit window offset register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x40	0x080	TRANSMIT_WINDOW_OFFSET	RW	Stores the transmit window offset 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_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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
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
Reserved	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
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	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
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 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 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. After firmware reads a packet, it decrements the counter by issuing the PACKET_RECEIVED command from the commander.	0
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.	0x631F

Field	Bit	Description	Reset

Table continues on the next page...

pdu_index_auto_updt	**15**	This bit enable the LL hardware to update the index of the PDU being queued in the data buffer. 0x0- LL Hardware to update the index of the PDU being queued. 0x1- LL firmware to update the index when PDU is queued.	0x0
mask_suto_at_update	14	This bit is used to enable/ disable masking of internal hardware supervision timeout trigger when switching from old connection parameters to new parameters. 1 - Enable 0 - Disable	0x01
extend_cu_tx_win	13	This bit is used to enable/ disable extending the additional rx window on slave side during connection update in event of packet miss at the update instant. 1 - Enable 0 - Disable	0x01
slv_md_config	12	Configuration is provided to send last data packet with MD bit set/zero or one (only for slave mode). SLV_MD_CONFIG bit has effect only when md_bit_ctr bit is not set 0 – The MD bit is set based on the transmit buffer empty condition. The last packet goes with a MD bit set since the FIFO would not be empty as we have to retain the data in the buffer till we realize the ACK from the remote side. 1 - MD bit will be controlled based on the availability of next data in the FIFO. So send last data packet with MD bit zero or one. If the data transmitted is the last packet in the FIFO the MD bit will be '0' in that packet. Only if the next location holds data then MD bit will be '1'.	0x0

Table continues on the next page...

aww_en	11	This field indicates whether window widen optimization is enable or not. 0 – It indicates AWW is enable 1 – It indicates AWW is disable	0x0
sl_dsm_en	10	This resiter field indicates whether slave latency dsm is enable or not. 0 – It indicates SL_DSM is enable 1 – It indicates SL_DSM is disable	0x0

index_not_in_addr	9	This register field indicates whether connection index is present in address or not. 1 – Index is not in address 0 – Index is in address	0x1
Sw_cntrl_md	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

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 LLH 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, LLH 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_expired_intr	15	If this bit is set ping timer nearly expired interrupt is enabled.	0
*ping_timer_expired_intr	14	If this bit is set ping timer expired interrupt is enabled.	0
Reserved	13:10	Unused.	0
rx_bad_pdu_int_en	9	If this bit is set packet receive bad pdu interrupt is enabled.	0

Table continues on the next page...

		Effective only when bit 6 is set.	
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

Table continues on the next page...

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 conn_config 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 window_widen_intvl register value to calculate the window widening size.	00A

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. Firmware shall update this register with proper index before writing/ reading the connection-specific registers (refer to register summary before for the connection specific register set). This register is relevant only for multiple connection support. If multiple connection is supported but number of connections is made one then the register value needs to be kept 0x0.	0x0000

Field	Bit	Description	Reset
RFU	15:3	Reserved	0
fw_conn_index	2:0	This field is used to index the multiple connections existing. Range is 0 to one less than maximum number of connections supported (limited to 4 as of now). 0x0 – Connection Engine 0 in LLH 0x1 – Connection Engine 1 in LLH	0

Note – An alternative for programming of the conn index register for every access to the connection registers is to pass the conn index along with the LLH address itself which requires address space to be available for this mapping.

*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 connection. To clear interrupt for a specific	0x0000

				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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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
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

Table continues on the next page...

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. This shall be a multiple of 1.25 ms in the range of 7.5 ms to 4.0 s.	0x0000

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

Table continues on the next page...

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	RO	This is the free running counter, connEventCounter as defined by Bluetooth spec.	0x0000

Field	Bit	Description	Reset
connectionEventCounter	15:0	Firmware will read the instantaneous Event counter from this register, during connection update and channel map update 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.	0x0000

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

Table continues on the next page...

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. Pause the data transfer on the connection. The current_pdu_index in hardware does not move to next index until pause_data is cleared.	0
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.	0

		Hardware will start the next data transmission from the index indicated by this field.	

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_new[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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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_latency_newl[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_so_to_new[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
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

Table continues on the next page...

sl_conn_interval_new[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 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.	0x000A

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	0x0000

				the time to get the response for the control procedures for which timeout rules are specified in the specification.	

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_resp_timer_expire	15:0	This field defines the clock instant at which the next PDU response timeout event will occur on a connection.	0x0000

		This is with reference to the 16-bit internal reference clock.	

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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	15:9	Unused	0
Set/Clear	8	Used to set the SENT bit in hardware for the selected packet buffer. 1 – packet queued	0

Table continues on the next page...

		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. Note: This register has a different meaning in the Read-path	
List_index	7:0	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	15:9	Unused	0x0000

Table continues on the next page...

Set/clear	8	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. 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	0
List_index	7:0	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

Table continues on the next page...

Reserved	15:8	Unused	0x0000
tx_sent	7:0	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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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	15:8	Unused	0x0000
tx_ack	7:0	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.	0

		Note : This register has a different meaning in the Write-path.	

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.

Device Data List Status Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x290	0x520	DEVICE_DATA_LIST_STATUS	RO	Status of data packets queued and sent.	0x0000

Field	Bit	Description	Reset
Reserved	15:8	Unused	0x0000
device_data_list_status	7:0	These bits the status of data pecked queued and sent at device level. If the particular bit is set to ‘1’ indicates that there either pending PDU in the queue or pending ACK to be cleared. Based on the status of this register FW will queue next PDU. Note : This register is used only in shared connection tx_fifo.	0

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
0x200 - 0x21E	0x400- 0x43C	DATA_MEM_DESCRIPTOR0 - DATA_MEM_DESCRIPTOR15	RW	Descriptor for packet stored in the each of the transmit buffer which includes the packet specific information like length and LLID.	0x0000

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.	0x000A

Field	Bit	Description	Reset
Reserved	7:2	RFU	0
llf_sl_dsm_ww_ctrl	3	Control bit to enable or disable the wakeup delay when slave latency is enabled.	1
slave_latency_en_sts	2	Read Only . Firmware reads this bit to know the status of slave latency enable.	0
llf_sl_dsm_en	1	LLF assisted slave latency dsm feature is enabled by setting this bit. When slave latency is enabled, this feature enables the slave to awaken out of DSM when it is necessary for the slave to receive a packet.	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

Adaptive Window Widening Config

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x50A	0xA14	AWW_CONFIG	RW	Enabling/Disabling of various AWW specific functions and AWW configurable parameters.	0xE244

Field	Bit	Description	Reset
aww_error_tolerence_en	15	This bit enables/disables the access code bit error tolerance functionality. 1 – enable 0 - disable	1
aww_tolerence_threshold	14:10	This value sets the lower bound on tolerance of bit errors in access code to overlook AWW history recapture. Eg., 0x18 implies 32-24 = 8 bit errors are tolerable.	0x18
aww_alg_auto_switch_en	9:9	This bit is used to enable/ disable the functionality of automatic switching between the AWW algorithms i.e The Conservative Mode and Greedy Mode. 1 – Enable 0 – Disable	1
aww_alg_sel	8:8	IF the AWW algorithm auto switch functionality is disabled, this bit is used to manually select the AWW algorithm to be used. 1 – Greedy Mode 0 – Conservative Mode	0
constancy_cnt_init_val	7:4	This value defines the number of connection events for which the adaptive window module checks if master anchor points variations fall within set threshold before switching from conservative mode to greedy mode. Valid Range : (0x1,0xf)	0x04
Obs_ce_cnt_init_val	3:0	This value defines the number of connection events for which the adaptive window module performs measurements of master anchor point swing before it applies the optimized window in accordance with conservative mode aww algorithm. Valid Range : (0x1,0xf)	0x04

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 calculated slave window widening value to be used as final window widen value.	0x10

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. Unit is 1.25ms	0x0000

Cumulative hop increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x25A	0x4B4	SL_CONN_HOP_INCREMENT	RW	This register holds the value of cumulative hop increment corresponding to the Slave latency.	0x00

Field	Bit	Description	Reset
Sl_conn_hop_increment	5:0	Firmware writes the cumulative hop increment value corresponding to the slave latency upon connection creation complete before enabling Slave latency. Range of the value is 0 to 36.	0x00

Cumulative window widen register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x25C	0x4B8	SL_CONN_WINDOW_WIDEN	RW	This register holds the value of cumulative window widen value corresponding to the slave latency.	0x0000

Field	Bit	Description	Reset
Sl_conn_window_widen	15:0	Firmware writes the cumulative window widen value corresponding to the slave latency upon connection creation complete before enabling Slave latency. Unit is in microseconds.	0x0000

Connection Update New slave latency hop increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x25E	0x4BC	CONN_UPDATE_NEW_SL_HOP_INCREMENT	RW	The (slaveLatency * hopIncrement) value that will be effective after the connection update instant.	0x00

Field	Bit	Description	Reset
Sl_hop_increment_new[5:0]	5:0	This register will have the new SL_New * hopIncrement value that the hardware will use after the connection update instant. Before the instant, the hop increment in the register SL_CONN_HOP_INCREMENT will be used by hardware. Range of the value is 0 to 36.	0x00

Connection Update New slave latency window widen register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x260	0x4C0	CONN_UPDATE_NEW_SL_WIN_WIDEN	RW	The cumulative window widen value corresponding to the new slave latency after the update instant.	0x0000

Field	Bit	Description	Reset
Sl_window_widen_new[15:0]	15:0	The cumulative window widen value after the connection update instant. Before the instant, the window widen in the register SL_CONN_WINDOW_WIDEN will be used by hardware. Unit is in microseconds.	0x0000

Previous connection instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x262	0x4C4	PREVIOUS_CONN_INSTANT	RO	Reference BT clock value of the previous anchor point.	0x0000

Field	Bit	Description	Reset
Previous_conn_instant	15:0	Read only register. Holds the value of the reference BT clock value at the previous connection event that LLH participated.	0x0000

Previous Connection instant event count

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x26E	0x4DC	PREVIOUS_CONN_CE_COUNT	RO	Connection event count of the previous anchor point.	0x0000

Field	Bit	Description	Reset
previous_conn_ce_count	15:0	Read only register. Holds the value of the event counter at the previous connection event that LLH participated.	0x0000

Previous Connection channel

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x270	0x4E0	PREVIOUS_CONN_CH	RO	Connection channel of the previous anchor point.	0x0000

Field	Bit	Description	Reset
previous_conn_ch	15:0	Read only register. Holds the value of the channel at the previous connection event that LLH participated.	0x0000

Slave latency connection event counter increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x264	0x4C8	CE_COUNT_INCREMENT_N	RW	This register holds the value of connection event count N.	0x0000

Field	Bit	Description	Reset

Table continues on the next page...

Ce_count_increment_n	15:0	Firmware programs the value of connection event count N corresponding to the elapsed time (from the previous anchor point) and expected next anchor point (anchor point when it wants to exit slave latency).	0x0000

Slave latency connection interval increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x266	0x4CC	CONN_INTERVAL_INCREMENT_N	RW	This register holds the value of cumulative connection interval for the connection event count N.	0x0000

Field	Bit	Description	Reset
Conn_interval_increment_n	15:0	Firmware writes the cumulative connection interval for the connection event count N corresponding to the elapsed time (from the previous anchor point) and expected next anchor point (anchor point when it wants to exit slave latency). Unit is 1.25ms	0x0000

Slave latency hop increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x268	0x4D0	CONN_HOP_INCREMENT_N	RW	This register holds the value of cumulative hop increment for the connection event count N.	0x00

Field	Bit	Description	Reset
Conn_hop_increment_n	5:0	Firmware writes the cumulative hop increment for the connection event count N corresponding to the elapsed time (from the previous	0x00

		anchor point) and expected next anchor point (anchor point when it wants to exit slave latency). Range of the value is 0 to 36.	

Slave latency window widen increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x26A	0x4D4	CONN_WINDOW_WIDEN_N	RW	This register holds the value of cumulative window widen for the connection event count N.	0x0000

Field	Bit	Description	Reset
Conn_window_widen_n	15:0	Firmware writes the cumulative window widen for the connection event count N corresponding to the elapsed time (from the previous anchor point) and expected next anchor point (anchor point when it wants to exit slave latency). Unit is in microseconds.	0x0000

Slave latency new connection instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x26C	0x4D8	ANCHOR_POINT_INCREMENT_N	RW	This register holds the value of the distance for connection event count N corresponding to the elapsed time.	0x0000

Field	Bit	Description	Reset
Anchor_point_increment_n	15:0	Firmware write the value of the distance for the connection event count N corresponding to the elapsed time from the previous anchor point. It calculate the distance as	0x0000

		(N * CI) – (((SL_CONN_WINDOW_WIDEN_N) slots) + 1)). Unit is 0.625ms LLH use this value to check the validity the instant passed for the new derived anchor point with reference to the PREVIOUS_INSTANT.	

**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).	0x0000

				This together with CONN_PING_TIMER_NEXT_EXP_WRAP will provide the correct status of ping timer duration.	

**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
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

#Connection Arbiter Parameter register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x102	0x204	CONN_ARB_PARAMS	RO	Register holds the highest priority request to the connection arbiter from the connection engines.	0xF

Field	Bit	Description	Reset
Reserved	15:4	Not used	--
priority_pointer	3:0	Pointer for the highest priority request to the connection arbiter.	0xF

Conn_SUTO_CI_Ratio register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x104	0x208	CONN_SUTO_CI_RATIO	RW	Firmware programs this register with connection supervision timeout to connection interval ratio at the time of connection creation/ connection update.	0x0000

Field	Bit	Description	Reset
RFU	15:13	Not used	0x0
conn_suto_ci_ratio[12:0]	12:0	This register will have the ratio of the connection supervision timeout to the connection interval as programmed by the firmware before a connection creation or a connection update. The value is used by hardware to update the priority of a connection link that is nearing connection timeout due to packet miss.	0x0000

Connection Priority Config register

Addr	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x106	0x20C	CONN_PRIORITY_CNFG	RW	This register is used by the firmware to override the hardware priority and set the priority of each connection link in the connection arbiter. The value of Zero has the least priority.	0x0000

Field	Bit	Description	Reset
priority_sel	15	If this bit is set to '1' connection arbiter will use the firmware programmed priority set in this register for each connection engine.	0
RFU	14:12	Not used	0x0
fw_priority3	11:9	Firmware programmed priority for connection engine 3.	0x0
fw_priority2	8:6	Firmware programmed priority for connection engine 2.	0x0
fw_priority1	5:3	Firmware programmed priority for connection engine 1.	0x0
fw_priority0	2:0	Firmware programmed priority for connection engine 0.	0x0

Connection Update New SUP_TO to CI ratio

Addr	Addr 32-bit	Register Name	RW	Description	Reset
0x1DA	0x3B4	CONN_UPDT_NEW_SU_CI_RATIO	RW	The Supervision Timeout to conn interval ratio that will be effective after the connection update instant. Firmware programs this register along with other connection update parameter registers.	0x0000

Field	Bit	Description	Reset
RFU	15:13	Not used.	0x0
conn_suto_ci_ratio_new	12:0	This register will have the new supervision timeout to connection interval ratio that the hardware will use after the connection update instant. Before the instant, the value in the register CONN_SUTO_CI_RATIO will be used by hardware.	0x0000

#Slave Window Offset Full register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset	
0x1DC	0x3B8	SL_WIN_OFF_FULL	RW	This register is programmed with the maximum possible window widen value by the firmware when dsm entry during slave latency feature is enabled. This value is used by the hardware to calculate the next ce instant being reported to firmware.	0x0000	

Field	Bit	Description	Reset
RFU	15:6	Not used.	0x0
sl_window_off_full	5:0	Value of maximum possible window widen programmed by the firmware when slave latency dsm feature is enabled.	0x0000

Connection RSSI register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset	
0x518	0xA30	CONN_RX_RSSI	RO	This connection register holds the RSSI value of the last good (empty/non-empty) packet received by the specific connection.	0x0000	

Field	Bit	Description	Reset

Table continues on the next page...

conn_rx_rssi	15:0	RSSI during the last good packet received in the specific connection.	0x0000

Connection Rx Memory read enable register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset	
0x522	0xA44	LLH_MEM_READ_ENABLE_CONTROL_REG	WO	This register is used by the firmware to control read access to the connection Rx memory.	0x0000	

Field	Bit	Description	Reset
Reserved	15:14	Not Used	0x0
conn_rxmem_rd_ctrl	0	When this bit is '1', connection Rx memory is enabled.	0

Tx memory configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x27A	0x4F4	TXMEM_CONFIG	RW	This register fields can be set to configure the size of the data PDUs.	0x700

Field	Bit	Description	Reset
RFU	15:12	Reserved	0x0
Number_of_packets_minus_1	11:9	Maximum Number of Data PDUs in Tx FIFO.	0x03
Max_data_pdu_size	8:0	Maximum data packet length.	0x100

Tx memory configuration register 2

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x27C	0x4F8	TXMEM_CONFIG_2	RW	This register fields can be set to configure the base address of the Control PDU.	0x300

Field	Bit	Description	Reset
RFU	15:11	Reserved	0x0
Max_total_data_size	10:0	Total size of Data PDUs. i.e. Maximum Number of Data PDUs * Maximum Data Packet Length as configured in TXMEM_CONFIG register. It is base address of first Control PDU.	0x300

45.5.1.1.4 Test and Debug Registers

Test and Debug Register Descriptions for the Bluetooth Link Layer

DTM control register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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
Reserved	15:10	Reserved for future use	0
dtm_conttx_en	13	Write: 1 – To enable continuous transmit mode 0 – To disable continuous transmit mode	0
Reserved	12:10	Not Used	
Pkt_payload[2:0]	9:7	Payload type as per the HCI parameter. 0x00 Pseudo-Random bit sequence 9	0

Table continues on the next page...

		0x01 Pattern of alternating bits ‘11110000 0x02 Pattern of alternating bits ‘10101010’ 0x03 Pseudo-Random bit sequence 15 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. Read: 1 – Indicates DTM test ON 0 – Indicates DTM test OFF Write: 1 – To enable continuous receive mode 0 – To disable continuous receive mode	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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

Table continues on the next page...

		0 – no change from normal behavior	
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
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.	0

Table continues on the next page...

		1 – corrupt CRC 0 – no change	
Rx_crc_err	2	Receive packet CRC error indication. 1 – Indicate CRC error of received packet, irrespective of good/bad CRC 0 – No change from normal behavior	0
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

DTM transmit packet count register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xC2	0x184	DTM_TX_PKT_COUNT	RO	Count of the number of LE packets transmitted when device is configured in transmit test mode.	0x0000

Field	Bit	Description	Reset
tx_packet_count[15:0]	15:0	Number of packets transmitted in transmit test mode. 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
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

DTM Length register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xCE	0x19C	LE_DTM_TEST_LEN	RW	LE Direct Test Mode (DTM) length register. Used to configure the payload length of direct test mode (DTM) operation.	0x0000

Field	Bit	Description	Reset
Reserved	15:8	Reserved for future use	0
dtm_len	7:0	0x00-0xFF (except 0x3F) Length in bytes of payload data in each packet 0x3F – Continuous Transmit mode.	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 Decimal	Output Sleep Frequency(KHz)	PPM	SCA	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.

DTM 2 wire UART *Baud rate configuration*

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x580	0xB00	DTM_2WIRE_CONFIG	RW	This value will go to different baud rate generator module for DTM 2 wire UART.	0x006F

Field	Bit	Description	Reset
dtm_2wire_baud_config	15:0	This value will be given to baud clock generator module. For different input clock frequency to LLH, this value will be different to generate corresponding baud clock as mentioned in below table.	0x006F

Radio clock	Baud rate	Configuration Value
26 MHz	1200	0x2A3B
	2400	0X151E

Table continues on the next page...

	9600	0X0546
	14400	0X0384
	19200	0X02A1
	38400	0X0152
	57600	0X00E1
	115200	0X006F
64 MHz	1200	0X19FA
	2400	0X0CFE
	9600	0X033E
	14400	0X0229
	19200	0X019E
	38400	0X00D0
	57600	0X008A
	115200	0X0042

Note: For lec_top (LLH + PHY) environment input frequency is 64 MHz. Value mentioned on above table for 64 MHz is w.r.t 16 MHz because we divide 64 MHz value by 4 internally. This 16 MHz derived value is given as an input to LLH and used for further calculation.

DTM Offset Configuration Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x118	0x230	DTM_OFFSET_CONFIG	RW	Contains configurable parameters for DTM function enhancement request.	0x012D

Field	Bit	Description	Reset
Reserved	15:11	Reserved for future use.	
dtm_offset_us_buffer	10:1	Represents TIFS value ‘150’ in the calculation of dtm offset.	0x96
dtm_offset_en	0	If set to 1, DTM offset enhancement is enabled. If set to 0, DTM offset enhancement is disabled.	0x1

-

DTM Transmit Packet Configuration Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x11A	0x234	DTM_TX_PKT_CONFIG	RW	Configures number of DTM packets to be transmitted when DTM Tx operation is triggered and automatically stops the DTM Tx operation after that many number of packets are transmitted.	0x0000

Field	Bit	Description	Reset
dtm_tx_pkt_config[15:0]	15:0	If this field value is non-zero, it represents number of DTM packets to be transmitted when dtm_start command is received. DTM Tx operation is automatically stopped when DTM Tx packet count equals to the configured value. If this field value is zero, DTM continues to transmit packets till dtm_stop command is issued.	0

DTM Expected Receive Packet Length Register

-

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x11C	0x238	DTM_EXP_RX_LEN	RW	Contains configurable parameters for DTM timeout feature.	0x0000

Field	Bit	Description	Reset
Reserved	15:9	Reserved for future use.	
dtm_exp_rx_length	8:1	Configured length value for DTM receiver.	0x00
dtm_exp_rx_len_en	0	If set to 1, DTM timeout feature is enabled. If set to 0, DTM timeout feature is disabled.	0x0

Packet Payload Status Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x11E	-	PKT_PAYLOAD_STATUS	RO	Contains CRC status and Payload length value of received packet.	0x0000

Field	Bit	Description	Reset
Reserved	15:9	Reserved for future use.	
crc_status	8	Contains the value of CRC status of the received packet. Updates each time when packet reception is over.	0x0
payload_length	7:0	Contains the value of payload length field. Updated each time when packet reception is over.	0x00

Packet RSSI Status Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x120	-	PKT_RSSI_STATUS	RO	Contains RSSI value for the received packet.	0x000

Field	Bit	Description	Reset
pkt_rssi_status	15:0	Contains the RSSI value of the received packet. Updated each time when packet reception is over.	0x0000

45.5.1.1.5 Interface Registers

Interface Register Descriptions for the Bluetooth Link Layer

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 timeout	0x0000

				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

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

RSSI Register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
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

RF_ACTIVE_PERIOD register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x5A	0x0B4	rf_active_period	RW	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: 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_polarity	6	Polarity bit for rf_active 0: active high	

Table continues on the next page...

		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

45.5.1.1.6 Instruction Set

Instruction Set Description for the Bluetooth Link Layer

Command	Opcode	Description
START_ADV	0x40	Start Advertiser operation. The associated Advertiser configuration registers are programmed before the command is issued.
STOP_ADV	0x41	Stop advertiser operation.
START_SCAN	0x42	Start scanner operation. The associated configuration registers must be programmed before the command is issued.
STOP_SCAN	0x43	Stop the scanner operation.
START_INIT	0x44	Start connection creation operation. The associated configuration registers must be programmed before the command is issued.
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.
DTM_RX_START	0x47	Start Direct Test Mode Receive Test. The associated configuration registers must be programmed before the command is issued.
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.

Table continues on the next page...

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.
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.
WL_CLK_ON	0x5d	Turn on clock to whitelist block
WL_CLK_OFF	0x5e	Turn off clock to whitelist block
UPDATE_CONN	0x68	Update connection parameters. Deprecated.
RC_ANCHOR_POINT	0x69	Instruct LLH to check the validity of the set and recalculate the new connection anchor point parameter if valid.
ENTER_AUTO_DSM	0x6A	Indicates CPU is idle.
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
ENTER_DSM_SHUTDOWN	0x77	Enter deep sleep mode with shutdown
STORE_START	0x78	Firmware triggers the store process
RESTORE_START	0x79	Firmware triggers the restore process
DATA_RESTORE_ON	0x7A	Firmware data restore enable
DATA_RESTORE_OFF	0x7B	Firmware data restore disable
START_RPA_TIMER	0x7C	Start Privacy RPA timer
STOP_RPA_TIMER	0x7D	Stop Privacy RPA timer
START_RPA_GEN	0x7E	Start RPA address generation
SOFT_RESET	0x80	Software reset. Resets all the hardware registers (except a few registers related to radio initialization).
LOAD_RAND_NUM	0x8C	Load random seed to LFSR for RPA generation

45.5.1.2 BTLE_RF register descriptions

45.5.1.2.1 BTLE_RF Memory map

BTLE_RF base address: 4005_B000h

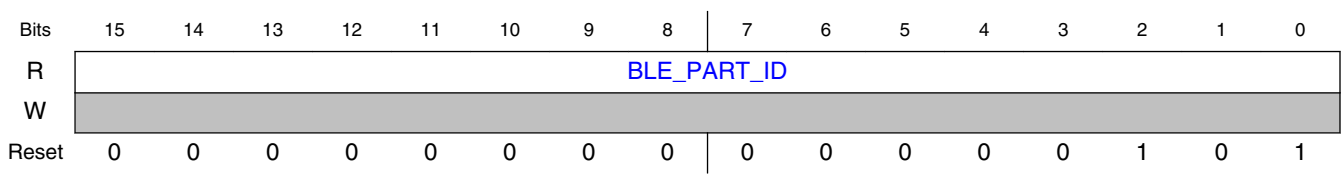
Offset	Register	Width (In bits)	Access	Reset value
600h	BLUETOOTH LOW ENERGY PART ID (BLE_PART_ID)	16	RO	0005h
604h	BLE DSM STATUS (DSM_STATUS)	16	RO	See description.
608h	BLE MISCELLANEOUS CONTROL (MISC_CTRL)	16	RW	0000h
60Ch	BLE STATE MACHINE STATUS (BLE_FSM)	16	RO	0000h

45.5.1.2.2 BLUETOOTH LOW ENERGY PART ID (BLE_PART_ID)

45.5.1.2.2.1 Offset

Register	Offset
BLE_PART_ID	600h

45.5.1.2.2.2 Diagram



45.5.1.2.2.3 Fields

Field	Function
15-0 BLE_PART_ID	BLE Part ID 0000000000000000b - Pre-production 0000000000000001b - Pre-production 0000000000000010b - KW40 0000000000000011b - KW41 0000000000000100b - K3S

Field	Function
	0000000000000101b - KW35/KW36

45.5.1.2.3 BLE DSM STATUS (DSM_STATUS)

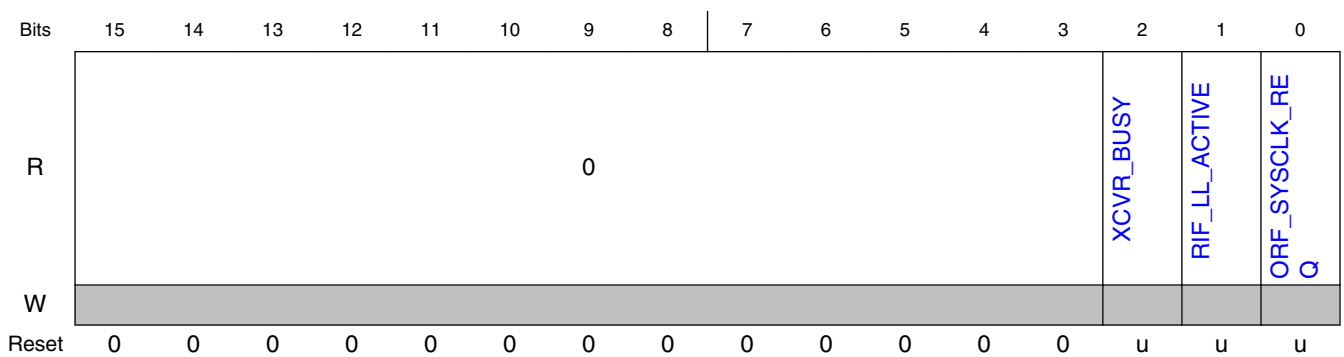
45.5.1.2.3.1 Offset

Register	Offset
DSM_STATUS	604h

45.5.1.2.3.2 Function

BLE Deep Sleep Mode Status Register

45.5.1.2.3.3 Diagram



45.5.1.2.3.4 Fields

Field	Function
15-3 —	Reserved.
2 XCVR_BUSY	Transceiver Busy Status Bit 0b - RF Channel in available (TSM is idle) 1b - RF Channel in use (TSM is busy)
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	RF Oscillator Requested

Field	Function
ORF_SYSCLK_REQ	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).

45.5.1.2.4 BLE MISCELLANEOUS CONTROL (MISC_CTRL)

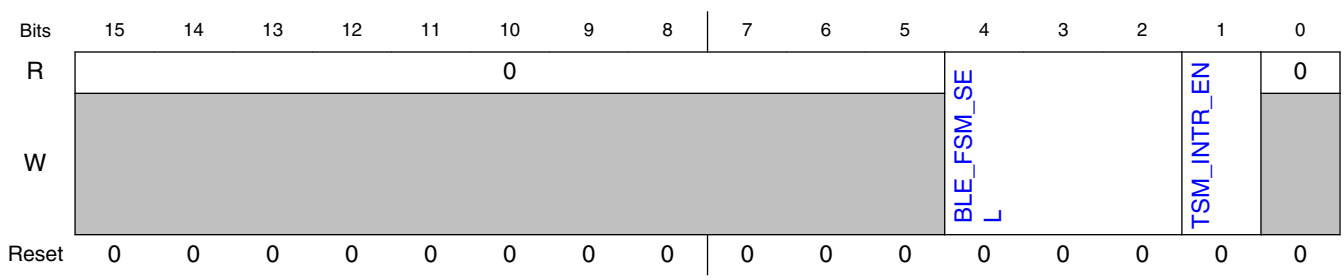
45.5.1.2.4.1 Offset

Register	Offset
MISC_CTRL	608h

45.5.1.2.4.2 Function

BLE Miscellaneous Control Register

45.5.1.2.4.3 Diagram



45.5.1.2.4.4 Fields

Field	Function
15-5 —	Reserved.
4-2 BLE_FSM_SEL	BLE FSM Selector BLE_FSM[2:0] selects which BLE state machine appears in the VAR_CS field of the BLE_FSM register, according to the following table: 0: VAR_CS = {2'b00, dtm_cs[2:0]} (DTM Current State) 1: VAR_CS = { adv_cs[4:0]} (ADV Current State) 2: VAR_CS = { scan_cs[4:0]} (SCAN Curernt State) 3: VAR_CS = { init_cs[4:0]} (INIT Current State) 4: VAR_CS = {1'b0, connection_cs[3:0]} (CONN Current State, using Connection Engine 0) 5: Reserved 6: Reserved 7: Reserved

Table continues on the next page...

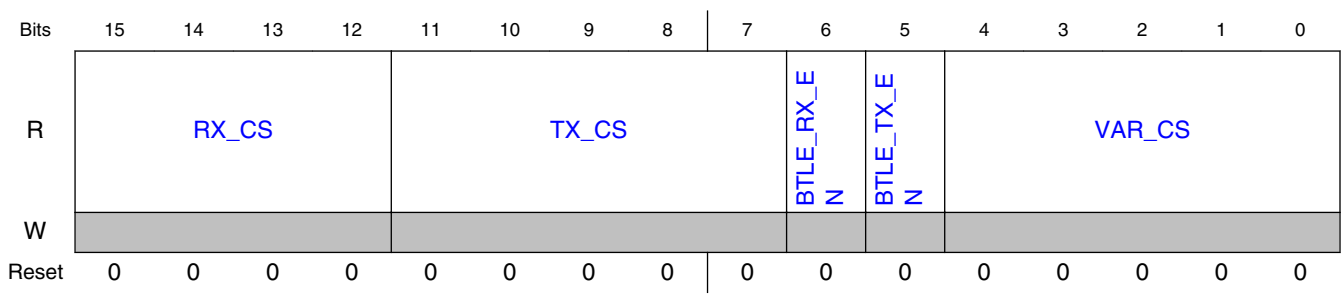
Field	Function
1 TSM_INTR_EN	TSM Interrupt Enable This control bit enables the TSM Interrupt for BLE. If TSM_INTR_EN=1 and a TSM Interrupt occurs during a BLE TX or RX operation, bit [9] of the BLE EVENT_STATUS register will become set.
0 —	Reserved.

45.5.1.2.5 BLE STATE MACHINE STATUS (BLE_FSM)

45.5.1.2.5.1 Offset

Register	Offset
BLE_FSM	60Ch

45.5.1.2.5.2 Diagram



45.5.1.2.5.3 Fields

Field	Function
15-12 RX_CS	BLE RX State Machine Current State Current State of *rx_cs[3:0]* (BLE RX state machine)
11-7 TX_CS	BLE TX State Machine Current State Current State of *tx_cs[4:0]* (BLE TX state machine)
6 BTLE_RX_EN	BLE RX Enable Control to TSM Current State of *btle_rx_en* (BLE TSM RX request input)
5 BTLE_TX_EN	BLE TX Enable Control to TSM Current State of *btle_tx_en* (BLE TSM TX request input)
4-0 VAR_CS	Variable State Machine Current State One of several BLE state machines can be mapped to VAR_CS[4:0]. See description of BLE_FSM_SEL[2:0] in the MISC_CTRL register

45.5.2 Generic_FSK Link Layer Controller

45.5.2.1 Introduction

The Generic FSK protocol enables radio operation using a custom GFSK/GMSK or MSK modulation format achieved by programming a set of PHY variables such as BT product, modulation index and modulation filter co-efficients (such that max frequency deviation <= 500kHz). Generic FSK mode also offers a highly configurable packet structure, variable bit rate transmission and reception, some limited packet (header) processing, and interface to a RAM-based Packet Buffer.

45.5.2.1.1 Overview

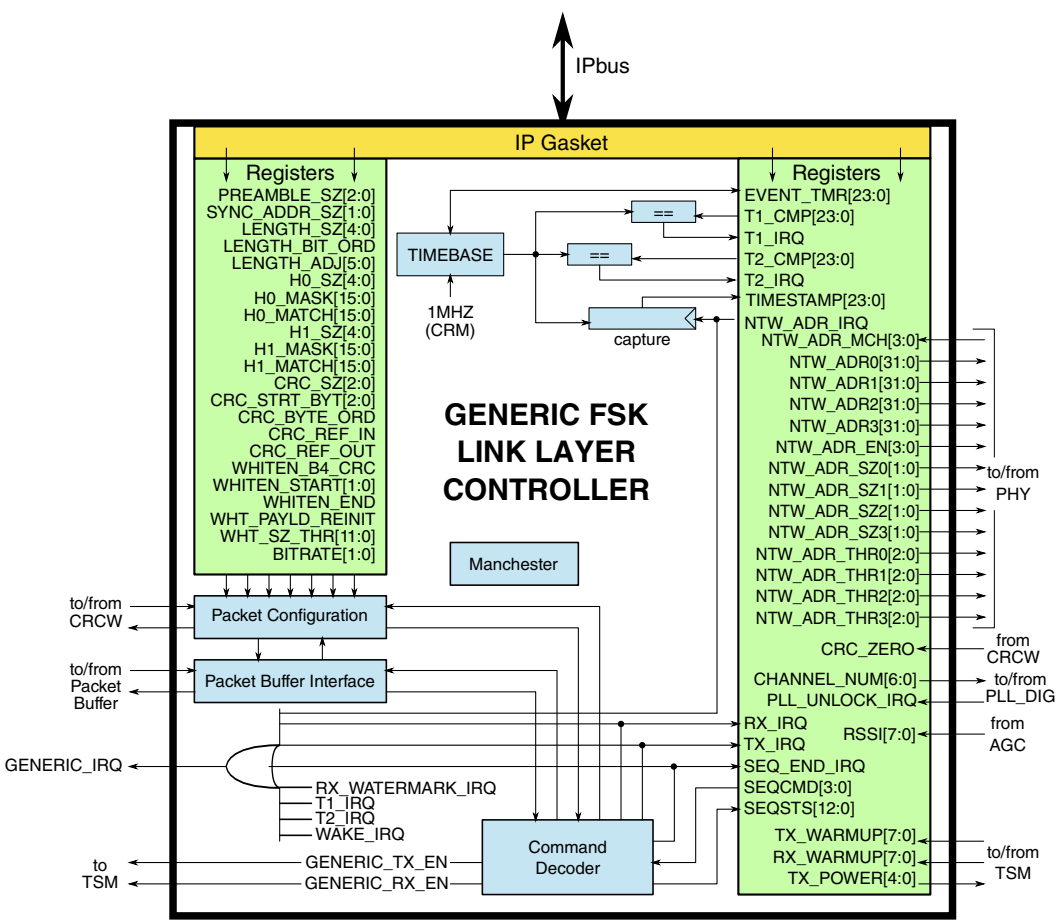
The Generic FSK Link Layer Controller provides the interface between Generic FSK software, and the 2.4GHz transceiver hardware. The interface consists of an IP bus-based set of registers, which provide command, control, and status to Generic FSK software. The Link Layer controller features programmability that allows for a highly configurable packet structure, defining the lengths, bit ordering, and contents of individual packet fields, defining the start and end points for whitening, CRC, and Manchester encoding/decoding, and providing for some primitive parsing of the packet header. The controller provides multiple options for bitrate, for both TX and RX. The Link Layer Controller implements a Sequence Command Set consisting of 12 commands; a command decoder to interpret the commands in realtime as they are received from Generic FSK software, and to provide the necessary control signals to the radio's Transceiver Sequence Manager (TSM); a high-precision timebase based on an external crystal-based oscillator; an Interrupt Control Unit; finite state machines to control the transmission and reception of Generic FSK packets; controls for CRC generation (TX) and verification (RX); controls for data whitening; Manchester encoding; interface to RAM-based Packet Buffer for packet storage; and provisions for entering and exiting low-power Deep Sleep Mode.

45.5.2.1.2 Features

- Highly configurable packet structure
- Optimized 12-entry Sequence Command Set
- High-precision timebase to maintain network timing

- Two timer-compare mechanisms for Interrupt Generation and Sequence Launching
- Hardware automation for packet transmit and receive, CRC, Whitening, and Manchester encoding
- Up to 4 Network Addresses to synchronize to, each can be 8-bit, 16-bit or 32-bit
- Packet Lengths up to 2047 Bytes
- Deep Sleep Mode support
- Highly configurable packet structure
- Optimized 12-entry Sequence Command Set
- High-precision timebase to maintain network timing
- Two timer-compare mechanisms for Interrupt Generation and Sequence Launching
- Hardware automation for packet transmit and receive, CRC, Whitening, and Manchester encoding
- Up to 4 Network Addresses to synchronize to, each can be 8-bit, 16-bit or 32-bit
- Packet Lengths up to 2047 Bytes
- Deep Sleep Mode support

45.5.2.1.3 Block Diagram



45.5.2.2 Memory Map and Register Definition

45.5.2.2.1 GENERIC FSK register descriptions

45.5.2.2.1.1 FSK Memory map

GENFSK base address: 4005_F000h

Offset	Register	Width (In bits)	Access	Reset value
0h	IRQ CONTROL (IRQ_CTRL)	32	RW	See description.
4h	EVENT TIMER (EVENT_TMR)	32	RW	See description.
8h	T1 COMPARE (T1_CMP)	32	RW	00FF_FFFFh
Ch	T2 COMPARE (T2_CMP)	32	RW	00FF_FFFFh
10h	TIMESTAMP (TIMESTAMP)	32	RO	See description.
14h	TRANSCEIVER CONTROL (XCVR_CTRL)	32	RW	See description.
18h	TRANSCEIVER STATUS (XCVR_STS)	32	RO	See description.
1Ch	TRANSCEIVER CONFIGURATION (XCVR_CFG)	32	RW	See description.
20h	CHANNEL NUMBER (CHANNEL_NUM)	32	RW	0000_0000h
24h	TRANSMIT POWER (TX_POWER)	32	RW	0000_0000h
28h	NETWORK ADDRESS CONTROL (NTW_ADR_CTRL)	32	RW	See description.
2Ch	NETWORK ADDRESS 0 (NTW_ADR_0)	32	RW	5555_5555h
30h	NETWORK ADDRESS 1 (NTW_ADR_1)	32	RW	5555_5555h
34h	NETWORK ADDRESS 2 (NTW_ADR_2)	32	RW	5555_5555h
38h	NETWORK ADDRESS 3 (NTW_ADR_3)	32	RW	5555_5555h
3Ch	RECEIVE WATERMARK (RX_WATERMARK)	32	RW	See description.
40h	DSM CONTROL (DSM_CTRL)	32	RW	0000_0000h
44h	PART ID (PART_ID)	32	RO	0000_0001h
60h	PACKET CONFIGURATION (PACKET_CFG)	32	RW	See description.
64h	H0 CONFIGURATION (H0_CFG)	32	RW	0000_0000h
68h	H1 CONFIGURATION (H1_CFG)	32	RW	0000_0000h

Table continues on the next page...

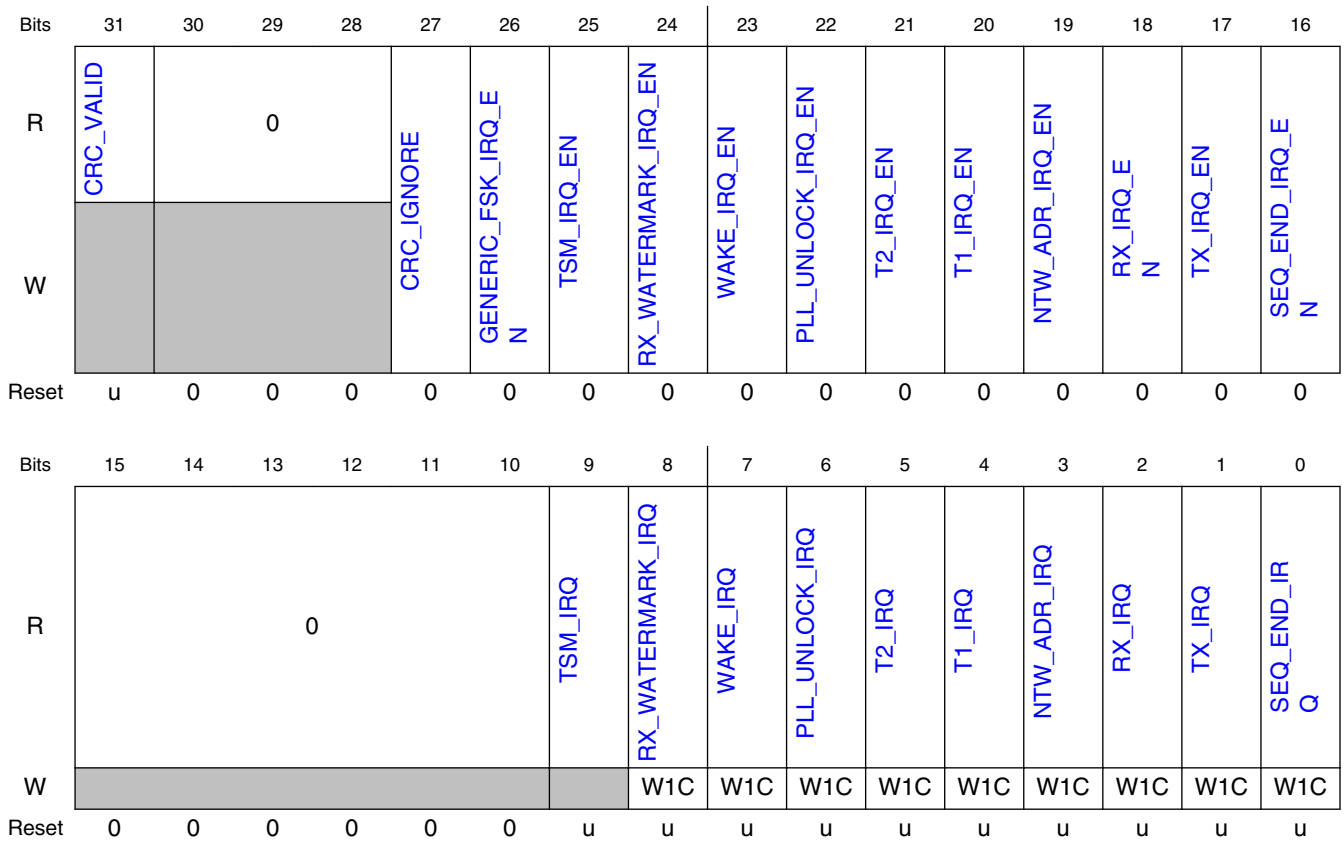
Offset	Register	Width (In bits)	Access	Reset value
6Ch	CRC CONFIGURATION (CRC_CFG)	32	RW	0000_0002h
70h	CRC INITIALIZATION (CRC_INIT)	32	RW	0000_0000h
74h	CRC POLYNOMIAL (CRC_POLY)	32	RW	1021_0000h
78h	CRC XOR OUT (CRC_XOR_OUT)	32	RW	0000_0000h
7Ch	WHITENER CONFIGURATION (WHITEN_CFG)	32	RW	01FF_0918h
80h	WHITENER POLYNOMIAL (WHITEN_POLY)	32	RW	0000_0021h
84h	WHITENER SIZE THRESHOLD (WHITEN_SZ_THR)	32	RW	0000_0800h
88h	BIT RATE (BITRATE)	32	RW	0000_0000h
8Ch	PACKET BUFFER PARTITION POINT (PB_PARTITION)	32	RW	0000_0220h
700h - F7Eh	PACKET BUFFER (PACKET_BUFFER_0 - PACKET_BUFFER_1087)	16	RW	See description.

45.5.2.2.1.2 IRQ CONTROL (IRQ_CTRL)

45.5.2.2.1.2.1 Offset

Register	Offset
IRQ_CTRL	0h

45.5.2.2.1.2.2 Diagram



45.5.2.2.1.2.3 Fields

Field	Function
31 CRC_VALID	CRC Valid CRC Valid indicator for RX packets. This bit becomes valid at RX_IRQ, and remains valid until the start of the next RX TSM sequence. 0b - CRC of RX packet is not valid. 1b - CRC of RX packet is valid.
30-28 —	Reserved.
27 CRC_IGNORE	CRC Ignore If set, assert RX_IRQ even for a received packet which fails CRC verification. 0b - RX_IRQ will not be asserted for a received packet which fails CRC verification. 1b - RX_IRQ will be asserted even for a received packet which fails CRC verification.
26 GENERIC_FSK_IRQ_EN	GENERIC_FSK_IRQ Master Enable Master enable for the GENERIC_FSK_IRQ interrupt line to the MCU. 0b - All GENERIC_FSK Interrupts are disabled. 1b - All GENERIC_FSK Interrupts can be enabled.
25	TSM_IRQ Enable 0b - TSM Interrupt is not enabled.

Table continues on the next page...

Field	Function
TSM_IRQ_EN	1b - TSM Interrupt is enabled.
24 RX_WATERMARK_IRQ_EN	RX_WATERMARK_IRQ Enable 0b - RX Watermark Interrupt is not enabled. 1b - RX Watermark Interrupt is enabled.
23 WAKE_IRQ_EN	WAKE_IRQ Enable 0b - Wake Interrupt is not enabled. 1b - Wake Interrupt is enabled.
22 PLL_UNLOCK_IRQ_EN	PLL_UNLOCK_IRQ Enable 0b - PLL Unlock Interrupt is not enabled. 1b - PLL Unlock Interrupt is enabled.
21 T2_IRQ_EN	T2_IRQ Enable 0b - Timer1 (T2) Compare Interrupt is not enabled. 1b - Timer1 (T2) Compare Interrupt is enabled.
20 T1_IRQ_EN	T1_IRQ Enable 0b - Timer1 (T1) Compare Interrupt is not enabled. 1b - Timer1 (T1) Compare Interrupt is enabled.
19 NTW_ADR_IRQ_EN	NTW_ADR_IRQ Enable 0b - Network Address Match Interrupt is not enabled. 1b - Network Address Match Interrupt is enabled.
18 RX_IRQ_EN	RX_IRQ Enable 0b - RX Interrupt is not enabled. 1b - RX Interrupt is enabled.
17 TX_IRQ_EN	TX_IRQ Enable 0b - TX Interrupt is not enabled. 1b - TX Interrupt is enabled.
16 SEQ_END_IRQ_EN	SEQ_END_IRQ Enable 0b - Sequence End Interrupt is not enabled. 1b - Sequence End Interrupt is enabled.
15-10 —	Reserved.
9 TSM_IRQ	TSM Interrupt Indicates TSM0_IRQ or TSM1_IRQ is set in XCVR_STATUS. Clear the bits there. For debug purposes. 0b - TSM0_IRQ and TSM1_IRQ are both clear. 1b - Indicates TSM0_IRQ or TSM1_IRQ is set in XCVR_STATUS.
8 RX_WATERMARK_IRQ	RX Watermark Interrupt Asserts when RX Byte Counter == RX_WATERMARK[12:0] 0b - RX Watermark Interrupt is not asserted. 1b - RX Watermark Interrupt is asserted.
7 WAKE_IRQ	Wake Interrrupt A WAKE_IRQ will be triggered when the GENERIC_FSK Link Layer Controller has awoken from a Manual DSM (Deep Sleep Mode) cycle. WAKE_IRQ indicates that the RF Oscillator has been restarted, and the GENERIC_FSK EVENT_TMR has resumed counting. 0b - Wake Interrupt is not asserted. 1b - Wake Interrupt is asserted.
6 PLL_UNLOCK_IRQ	PLL Unlock Interrupt An unlock event has occurred. 0b - PLL Unlock Interrupt is not asserted.

Table continues on the next page...

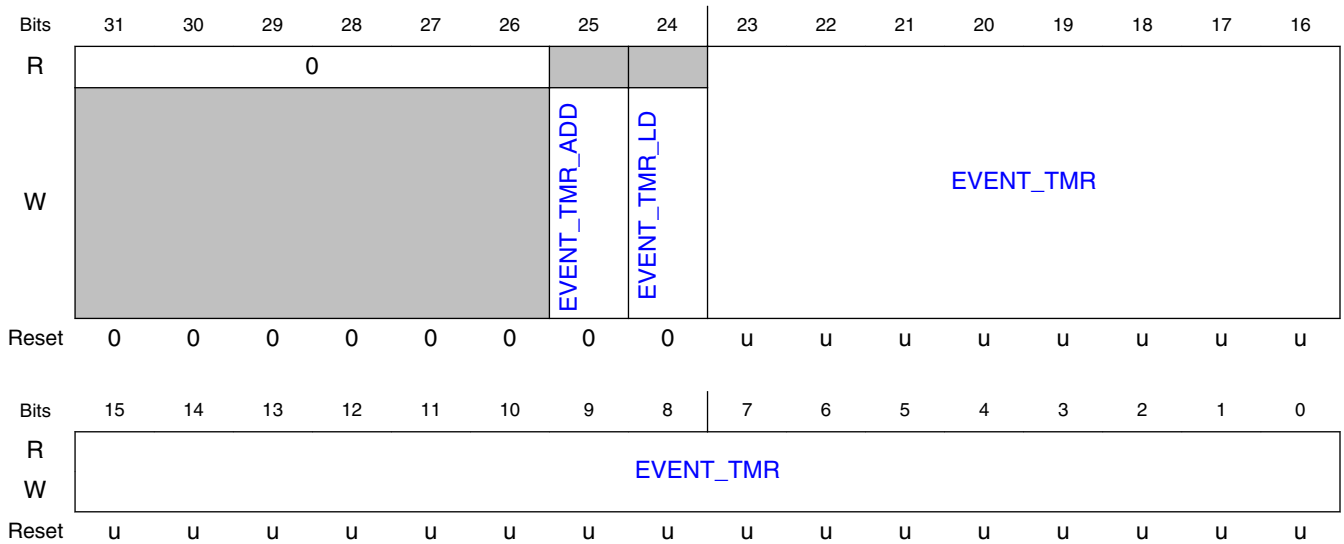
Field	Function
	1b - PLL Unlock Interrupt is asserted.
5 T2_IRQ	Timer2 (T2) Compare Interrupt 0b - Timer2 (T2) Compare Interrupt is not asserted. 1b - Timer2 (T2) Compare Interrupt is asserted.
4 T1_IRQ	Timer1 (T1) Compare Interrupt 0b - Timer1 (T1) Compare Interrupt is not asserted. 1b - Timer1 (T1) Compare Interrupt is asserted.
3 NTW_ADR_IRQ	Network Address Match Interrupt A Network Address Match has occurred. 0b - Network Address Match Interrupt is not asserted. 1b - Network Address Match Interrupt is asserted.
2 RX_IRQ	RX Interrupt The RX sequence has completed with a successful packet reception. 0b - RX Interrupt is not asserted. 1b - RX Interrupt is asserted.
1 TX_IRQ	TX Interrupt The TX sequence has completed with a successful packet transmission. 0b - TX Interrupt is not asserted. 1b - TX Interrupt is asserted.
0 SEQ_END_IRQ	Sequence End Interrupt Will assert when any TX or RX sequence ends for any reason. 0b - Sequence End Interrupt is not asserted. 1b - Sequence End Interrupt is asserted.

45.5.2.2.1.3 EVENT TIMER (EVENT_TMR)

45.5.2.2.1.3.1 Offset

Register	Offset
EVENT_TMR	4h

45.5.2.2.1.3.2 Diagram



45.5.2.2.1.3.3 Fields

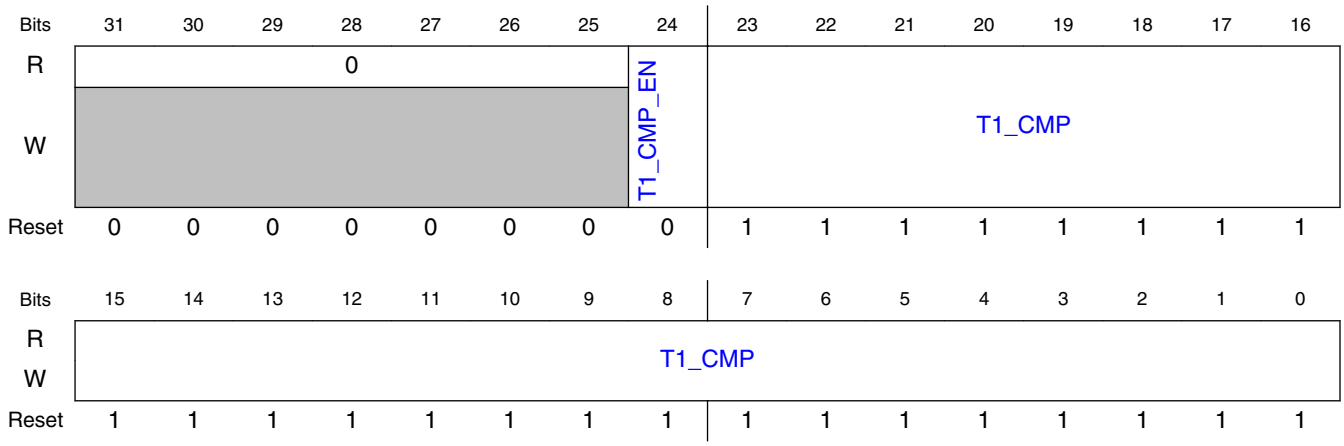
Field	Function
31-26 —	Reserved.
25 EVENT_TMR_ADD	Event Timer Add A write access with this bit increments EVENT_TMR by the contents of EVENT_TMR[23:0]. This is a signed addition.
24 EVENT_TMR_LD	Event Timer Load A write access with this bit set loads EVENT_TMR with the contents of EVENT_TMR[23:0]
23-0 EVENT_TMR	Event Timer Event Timer can be read in these byte locations. To update the Event Timer, either: 1. Write the desired EVENT_TMR to these bytes and set EVENT_TMR_LD=1, or, 2. Write the desired EVENT_TMR increment amount to these bytes and set EVENT_TMR_ADD=1. **Note:** for EVENT_TMR_ADD, EVENT_TMR[23:0] is a signed, two's-complement value.

45.5.2.2.1.4 T1 COMPARE (T1_CMP)

45.5.2.2.1.4.1 Offset

Register	Offset
T1_CMP	8h

45.5.2.2.1.4.2 Diagram



45.5.2.2.1.4.3 Fields

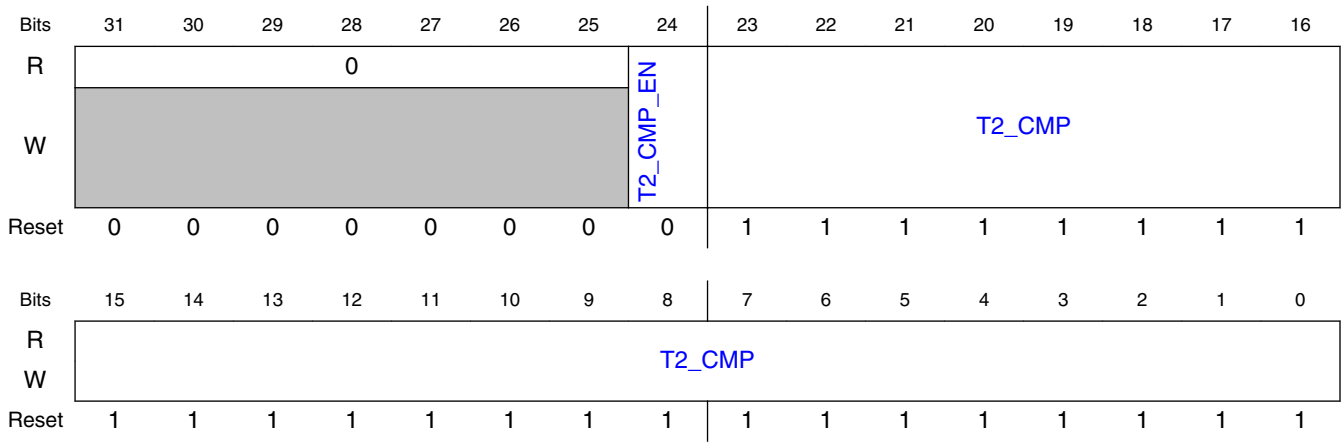
Field	Function
31-25 —	Reserved.
24 T1_CMP_EN	Timer1 (T1) Compare Enable Enable Timer Compare #1 (T1_CMP) to generate T1_IRQ and/or execute Sequence Commands.
23-0 T1_CMP	Timer1 (T1) Compare Value Timer1 (T1) Compare Value. Can be used to generate T1_IRQ and/or launch Sequence Commands

45.5.2.2.1.5 T2 COMPARE (T2_CMP)

45.5.2.2.1.5.1 Offset

Register	Offset
T2_CMP	Ch

45.5.2.2.1.5.2 Diagram



45.5.2.2.1.5.3 Fields

Field	Function
31-25 —	Reserved.
24 T2_CMP_EN	Timer2 (T2) Compare Enable Enable Timer Compare #2 (T2_CMP) to generate T2_IRQ and/or execute Sequence Commands.
23-0 T2_CMP	Timer2 (T2) Compare Value Timer2 (T2) Compare Value. Can be used to generate T2_IRQ and/or launch Sequence Commands

45.5.2.2.1.6 TIMESTAMP (TIMESTAMP)

45.5.2.2.1.6.1 Offset

Register	Offset
TIMESTAMP	10h

45.5.2.2.1.6.2 Diagram

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

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

45.5.2.2.1.6.3 Fields

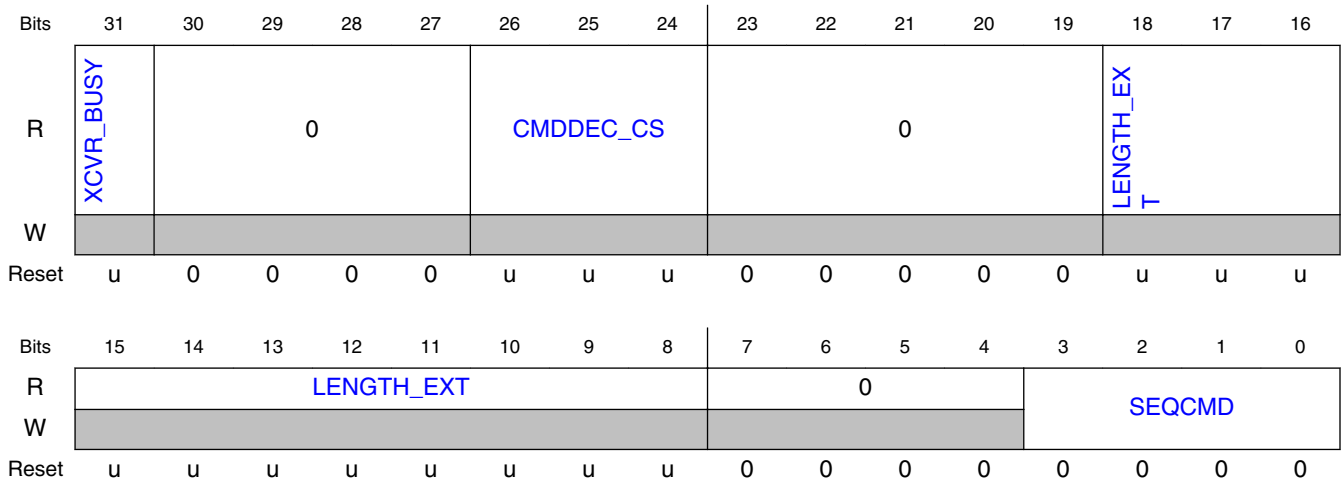
Field	Function
31-24 —	Reserved.
23-0 TIMESTAMP	Received Packet Timestamp Received Packet Timestamp. Captured at NTW_ADR_IRQ.

45.5.2.2.1.7 TRANSCEIVER CONTROL (XCVR_CTRL)

45.5.2.2.1.7.1 Offset

Register	Offset
XCVR_CTRL	14h

45.5.2.2.1.7.2 Diagram



45.5.2.2.1.7.3 Fields

Field	Function
31 XCVR_BUSY	Transceiver Busy For multi-protocol arbitration, XCVR_BUSY=1 indicates an RX or TX operation is underway, by either GENERIC_FSK, or some other protocol. 0b - IDLE 1b - BUSY
30-27 —	Reserved.
26-24 CMDDEC_CS	Command Decode Current State of the Command Decoder FSM (debug only)
23-19 —	Reserved.
18-8 LENGTH_EXT	Extracted Length Field The read-only register reflects the contents of the LENGTH field of the Header of the most-recently received packet. This is the raw, unmodified LENGTH field, as it appears in the received bit stream, unmodified by LENGTH_ADJ or any other parameter.
7-4 —	Reserved.
3-0 SEQCMD	Sequence Commands 0000b - No Action 0001b - TX Start Now 0010b - TX Start @ T1 Timer Compare Match (EVENT_TMR = T1_CMP) 0011b - TX Start @ T2 Timer Compare Match (EVENT_TMR = T2_CMP) 0100b - TX Cancel -- Cancels pending TX events but do not abort a TX-in-progress 0101b - RX Start Now 0110b - RX Start @ T1 Timer Compare Match (EVENT_TMR = T1_CMP) 0111b - RX Start @ T2 Timer Compare Match (EVENT_TMR = T2_CMP) 1000b - RX Stop @ T1 Timer Compare Match (EVENT_TMR = T1_CMP)

Field	Function
	1001b - RX Stop @ T2 Timer Compare Match (EVENT_TMR = T2_CMP) 1010b - RX Cancel -- Cancels pending RX events but do not abort a RX-in-progress 1011b - Abort All - Cancels all pending events and abort any sequence-in-progress 1100b - Reserved 1101b - Reserved 1110b - Reserved 1111b - Reserved

45.5.2.2.1.8 TRANSCEIVER STATUS (XCVR_STS)

45.5.2.2.1.8.1 Offset

Register	Offset
XCVR_STS	18h

45.5.2.2.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LQI								RSSI							
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CRC_VALID	LQI_VALID	0	RX_IN_WARMDN	RX_IN_PROGRESS	RX_IN_SEARCH	RX_IN_WARMUP	RX_STOP_T2_PEND	RX_STOP_T1_PEND	RX_START_T2_PEND	RX_START_T1_PEND	TX_IN_WARMDN	TX_IN_PROGRESS	TX_IN_WARMUP	TX_START_T2_PEND	TX_START_T1_PEND
W																
Reset	u	u	0	u	u	u	u	u	u	u	u	u	u	u	u	u

45.5.2.2.1.8.3 Fields

Field	Function
31-24 LQI	Link Quality Indicator This field is valid when LQI_VALID=1. LQI is a unsigned, unitless value.
23-16	Received Signal Stength Indicator, in dBm

Table continues on the next page...

Field	Function
RSSI	
15 CRC_VALID	CRC Valid Indicator CRC Valid indicator for RX packets. 0b - CRC is not valid for RX packet. 1b - CRC is valid for RX packet.
14 LQI_VALID	LQI Valid Indicator LQI Valid indicator for RX packets. This bit becomes set when the LQI computation completes for the packet currently being received, and remains set for the remainder of the packet. 0b - LQI is not yet valid for RX packet. 1b - LQI is valid for RX packet.
13 —	Reserved.
12 RX_IN_WARMDN	RX Warmdown Status RX Sequence in TSM Warmdown
11 RX_IN_PROGRESS	RX in Progress Status RX Packet Reception Currently Underway
10 RX_IN_SEARCH	RX Search Status RX Sequence in Network Address Search
9 RX_IN_WARMUP	RX Warmup Status RX Sequence in TSM Warmup
8 RX_STOP_T2_PEND	RX T2 Start Pending Status RX Sequence will stop @ next T2 Match
7 RX_STOP_T1_PEND	RX T1 Stop Pending Status RX Sequence will stop @ next T1 Match
6 RX_START_T2_PEND	RX T2 Start Pending Status RX Sequence will start @ next T2 Match
5 RX_START_T1_PEND	RX T1 Start Pending Status RX Sequence will start @ next T1 Match
4 TX_IN_WARMDN	TX Warmdown Status TX Sequence in TSM Warmdown
3 TX_IN_PROGRESS	TX in Progress Status TX Packet Transmission Currently Underway
2 TX_IN_WARMUP	TX Warmup Status TX Sequence in TSM Warmup

Table continues on the next page...

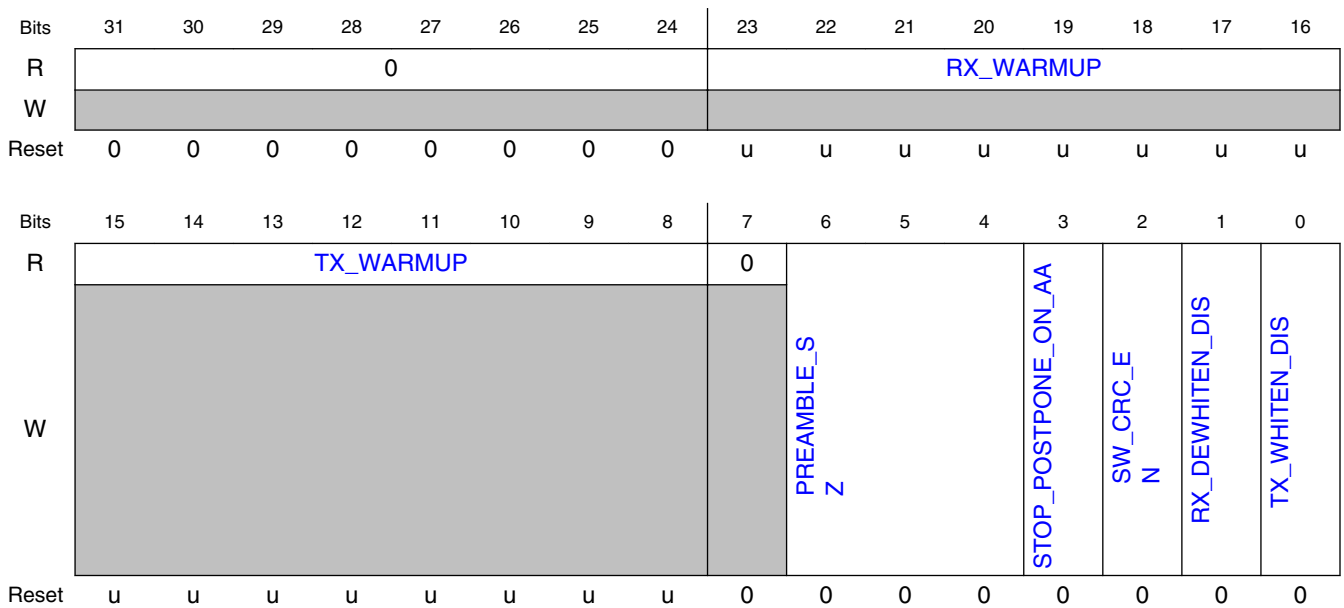
Field	Function
1 TX_START_T2_PEND	TX T2 Start Pending Status TX Sequence will start @ next T2 Match
0 TX_START_T1_PEND	TX T1 Start Pending Status TX Sequence will start @ next T1 Match

45.5.2.2.1.9 TRANSCEIVER CONFIGURATION (XCVR_CFG)

45.5.2.2.1.9.1 Offset

Register	Offset
XCVR_CFG	1Ch

45.5.2.2.1.9.2 Diagram



45.5.2.2.1.9.3 Fields

Field	Function
31-24 —	Reserved.

Table continues on the next page...

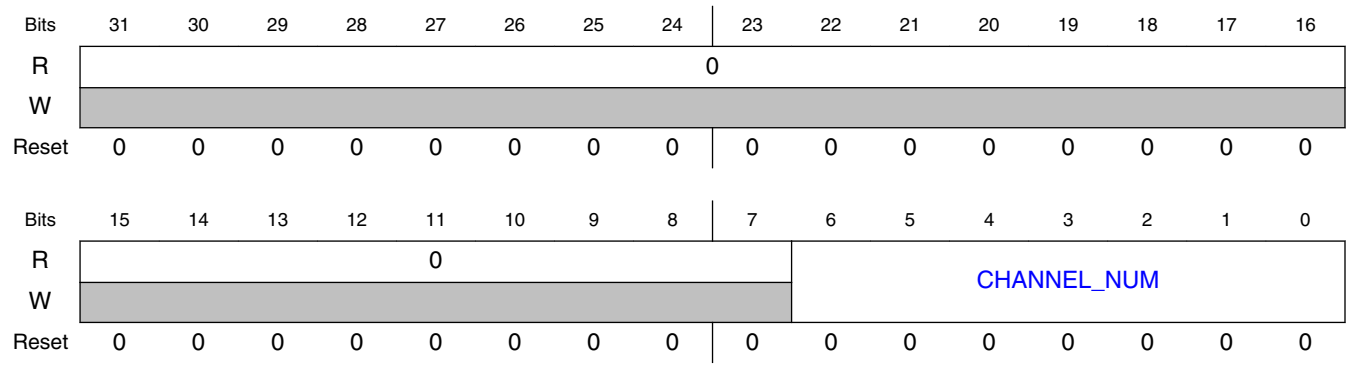
Field	Function
23-16 RX_WARMUP	Receive Warmup Time RX warmup time, in microseconds, provided to Link Layer software, so that warmup time can be accounted for when scheduling over-the-air transactions.
15-8 TX_WARMUP	Transmit Warmup Time TX warmup time, in microseconds, provided to Link Layer software, so that warmup time can be accounted for when scheduling over-the-air transactions.
7 —	Reserved.
6-4 PREAMBLE_SZ	Preamble Size Number of Octets = PREAMBLE_SZ + 1, where 0 <= PREAMBLE_SZ <= 7
3 STOP_POSTPONE_ON_AA	Postpone Stop Command Timeout On Access Address Match Enable If this bit is set, if a RX STOP condition occurs, due to an Event Timer match to a previously issued RX_STOP_T1 or RX_STOP_T2 sequence command, while a packet is being received (*ntw_adr_matched* = 1), the timeout-related abort will be deferred untl the packet reception is completed. If the packet is good, HW will signal Data Indication; otherwise the Sequence Manager will return to IDLE and await further commanding. 0b - STOP Abort will occur on RX_STOP_T1 or RX_STOP_T1 Event Timer match, regardless of ntw_adr_matched 1b - STOP Abort will be deferred on RX_STOP_T1 or RX_STOP_T1 Event Timer match, if ntw_adr_matched is asserted; otherwise the RX_STOP Abort will occur immediately
2 SW_CRC_EN	Software CRC Enable Software override of the HW-computed CRC for TX. Software must write CRC to Packet Buffer (RAM)
1 RX_DEWHITEN_DIS	RX De-Whitening Disable Disable all de-whitening on RX packets
0 TX_WHITEN_DIS	TX Whitening Disable Disable all whitening on TX packets

45.5.2.2.1.10 CHANNEL NUMBER (CHANNEL_NUM)

45.5.2.2.1.10.1 Offset

Register	Offset
CHANNEL_NUM	20h

45.5.2.2.1.10.2 Diagram



45.5.2.2.1.10.3 Fields

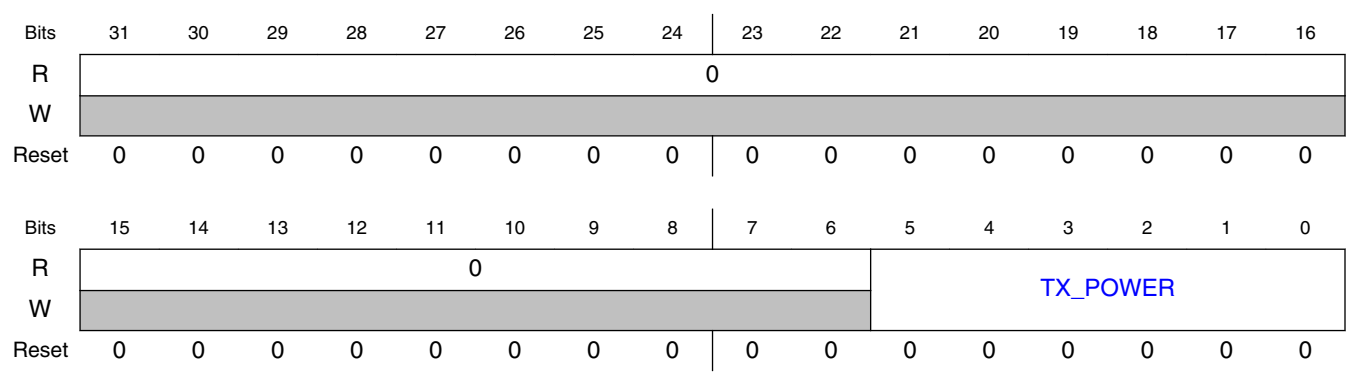
Field	Function
31-7 —	Reserved.
6-0 CHANNEL_NUM	Channel Number RF Channel Select: 0 <= CHANNEL_NUM <= 127; Formula: F = (2360 + CHANNEL_NUM) [in MHz]

45.5.2.2.1.11 TRANSMIT POWER (TX_POWER)

45.5.2.2.1.11.1 Offset

Register	Offset
TX_POWER	24h

45.5.2.2.1.11.2 Diagram



45.5.2.2.1.11.3 Fields

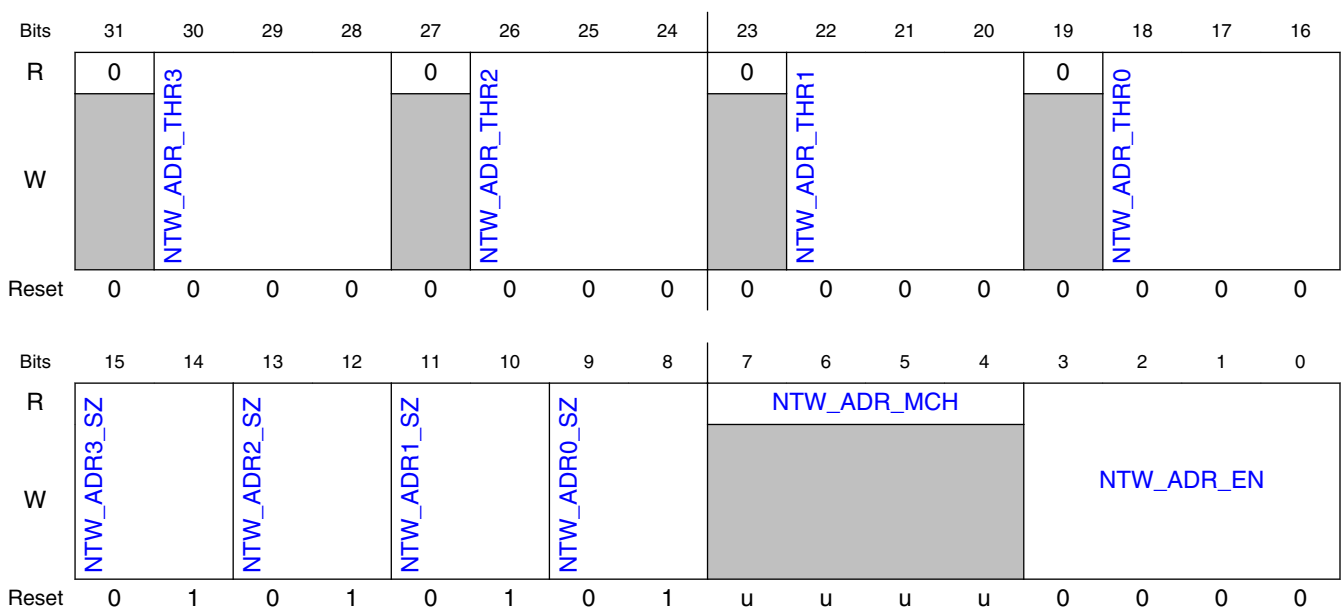
Field	Function
31-6 —	Reserved.
5-0 TX_POWER	Transmit Power PA Power Level.

45.5.2.2.1.12 NETWORK ADDRESS CONTROL (NTW_ADR_CTRL)

45.5.2.2.1.12.1 Offset

Register	Offset
NTW_ADR_CTRL	28h

45.5.2.2.1.12.2 Diagram



45.5.2.2.1.12.3 Fields

Field	Function
31	Reserved.

Table continues on the next page...

Field	Function
—	
30-28 NTW_ADR_THR3	Network Address 3 Threshold Number of Tolerated bit errors for Network Address 3
27 —	Reserved.
26-24 NTW_ADR_THR2	Network Address 2 Threshold Number of Tolerated bit errors for Network Address 2
23 —	Reserved.
22-20 NTW_ADR_THR1	Network Address 1 Threshold Number of Tolerated bit errors for Network Address 1
19 —	Reserved.
18-16 NTW_ADR_THR0	Network Address 0 Threshold Number of Tolerated bit errors for Network Address 0
15-14 NTW_ADR3_SZ	Network Address 3 Size 00b - Network Address 3 requires a 8-bit correlation 01b - Network Address 3 requires a 16-bit correlation 10b - Network Address 3 requires a 24-bit correlation 11b - Network Address 3 requires a 32-bit correlation
13-12 NTW_ADR2_SZ	Network Address 2 Size 00b - Network Address 2 requires a 8-bit correlation 01b - Network Address 2 requires a 16-bit correlation 10b - Network Address 2 requires a 24-bit correlation 11b - Network Address 2 requires a 32-bit correlation
11-10 NTW_ADR1_SZ	Network Address 1 Size 00b - Network Address 1 requires a 8-bit correlation 01b - Network Address 1 requires a 16-bit correlation 10b - Network Address 1 requires a 24-bit correlation 11b - Network Address 1 requires a 32-bit correlation
9-8 NTW_ADR0_SZ	Network Address 0 Size 00b - Network Address 0 requires a 8-bit correlation 01b - Network Address 0 requires a 16-bit correlation 10b - Network Address 0 requires a 24-bit correlation 11b - Network Address 0 requires a 32-bit correlation
7-4 NTW_ADR_MCH	Network Address Match Indicates which of the 4 Network Addresses has matched in the PHY. Valid during an RX sequence at the point of match, and remains asserted until either: 1. The next RX sequence begins (if the current packet passed CRC and header filtering), or, 2. An RX recycle to Network Address search (if the current packet failed CRC or header filtering) 0001b - Network Address 0 has matched 0010b - Network Address 1 has matched

Table continues on the next page...

Field	Function
	0100b - Network Address 2 has matched 1000b - Network Address 3 has matched
3-0 NTW_ADR_EN	Network Address Enable Enable Network Address N for PHY correlation, where 0 ≤ N ≤ 3. Any bit combination can be set. 0001b - Enable Network Address 0 for correlation 0010b - Enable Network Address 1 for correlation 0100b - Enable Network Address 2 for correlation 1000b - Enable Network Address 3 for correlation

45.5.2.2.1.13 NETWORK ADDRESS 0 (NTW_ADR_0)

45.5.2.2.1.13.1 Offset

Register	Offset
NTW_ADR_0	2Ch

45.5.2.2.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NTW_ADR_0															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NTW_ADR_0															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

45.5.2.2.1.13.3 Fields

Field	Function
31-0 NTW_ADR_0	Network Address 0 The PHY will search for this Network Address if NTW_ADR_CTRL[NTW_ADR_EN[0]] = 1

45.5.2.2.1.14 NETWORK ADDRESS 1 (NTW_ADR_1)

45.5.2.2.1.14.1 Offset

Register	Offset
NTW_ADR_1	30h

45.5.2.2.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NTW_ADR_1															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NTW_ADR_1															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

45.5.2.2.1.14.3 Fields

Field	Function
31-0 NTW_ADR_1	Network Address 1 The PHY will search for this Network Address if NTW_ADR_CTRL[NTW_ADR_EN[1]] = 1

45.5.2.2.1.15 NETWORK ADDRESS 2 (NTW_ADR_2)

45.5.2.2.1.15.1 Offset

Register	Offset
NTW_ADR_2	34h

45.5.2.2.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NTW_ADR_2															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NTW_ADR_2															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

45.5.2.2.1.15.3 Fields

Field	Function
31-0 NTW_ADR_2	Network Address 2 The PHY will search for this Network Address if NTW_ADR_CTRL[NTW_ADR_EN[2]] = 1

45.5.2.2.1.16 NETWORK ADDRESS 3 (NTW_ADR_3)

45.5.2.2.1.16.1 Offset

Register	Offset
NTW_ADR_3	38h

45.5.2.2.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NTW_ADR_3															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NTW_ADR_3															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

45.5.2.2.1.16.3 Fields

Field	Function
31-0 NTW_ADR_3	Network Address 2 The PHY will search for this Network Address if NTW_ADR_CTRL[NTW_ADR_EN[3]] = 1

45.5.2.2.1.17 RECEIVE WATERMARK (RX_WATERMARK)

45.5.2.2.1.17.1 Offset

Register	Offset
RX_WATERMARK	3Ch

45.5.2.2.1.17.2 Diagram

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

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

45.5.2.2.1.17.3 Fields

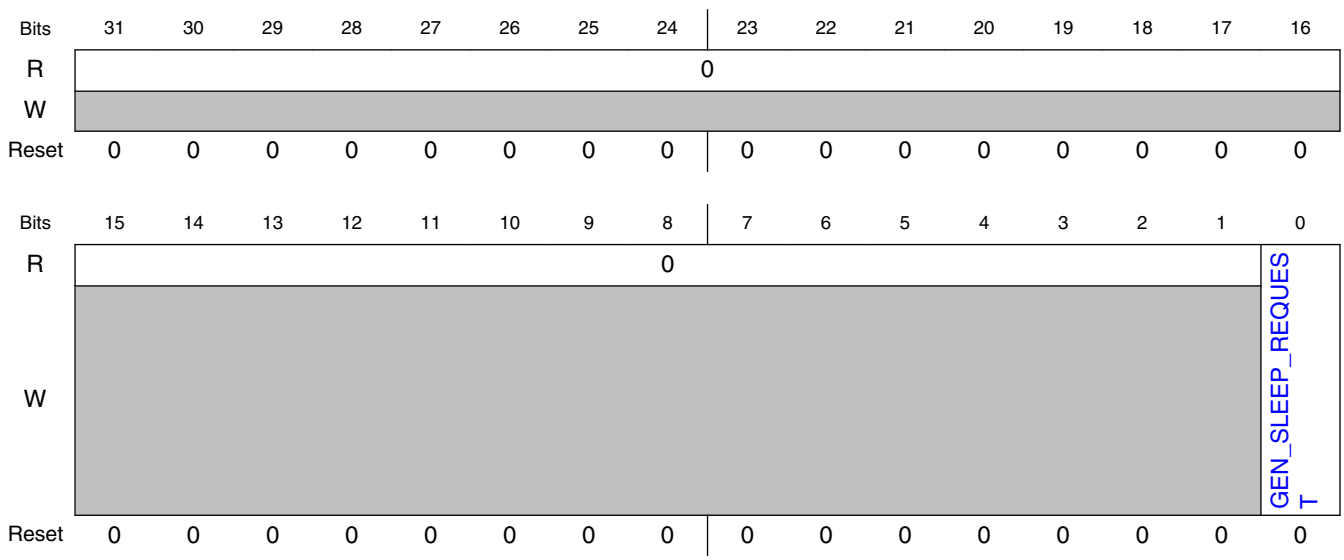
Field	Function
31-29 —	Reserved.
28-16 BYTE_COUNTER	Byte Counter Reflects the current Byte Count, for TX and RX. This is a signed, twos-complement value. For values less than zero, indicates the preamble transmission is underway (TX only). A value of 0 indicates the first octet of Network Address is being transmitted or received. A value of 1 indicates the second octet of Network Address is being transmitted or received. Etc.
15-13 —	Reserved.
12-0 RX_WATERMARK	Receive Watermark Sets the trigger for RX_WATERMARK_IRQ. Trigger the RX_WATERMARK_IRQ when: RX Byte Counter == RX_WATERMARK[12:0]

45.5.2.2.1.18 DSM CONTROL (DSM_CTRL)

45.5.2.2.1.18.1 Offset

Register	Offset
DSM_CTRL	40h

45.5.2.2.1.18.2 Diagram



45.5.2.2.1.18.3 Fields

Field	Function
31-1 —	Reserved.
0 GEN_SLEEP_REQUEST	GENERIC_FSK Deep Sleep Mode Request Setting GEN_SLEEP_REQUEST to 1 enables Deep Sleep Mode (DSM) to be entered when the RSIM DSM_TIMER matches the RSIM MAN_SLEEP register.

45.5.2.2.1.19 PART ID (PART_ID)

45.5.2.2.1.19.1 Offset

Register	Offset
PART_ID	44h

45.5.2.2.1.19.2 Diagram

Bits	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

Bits	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	1

45.5.2.2.1.19.3 Fields

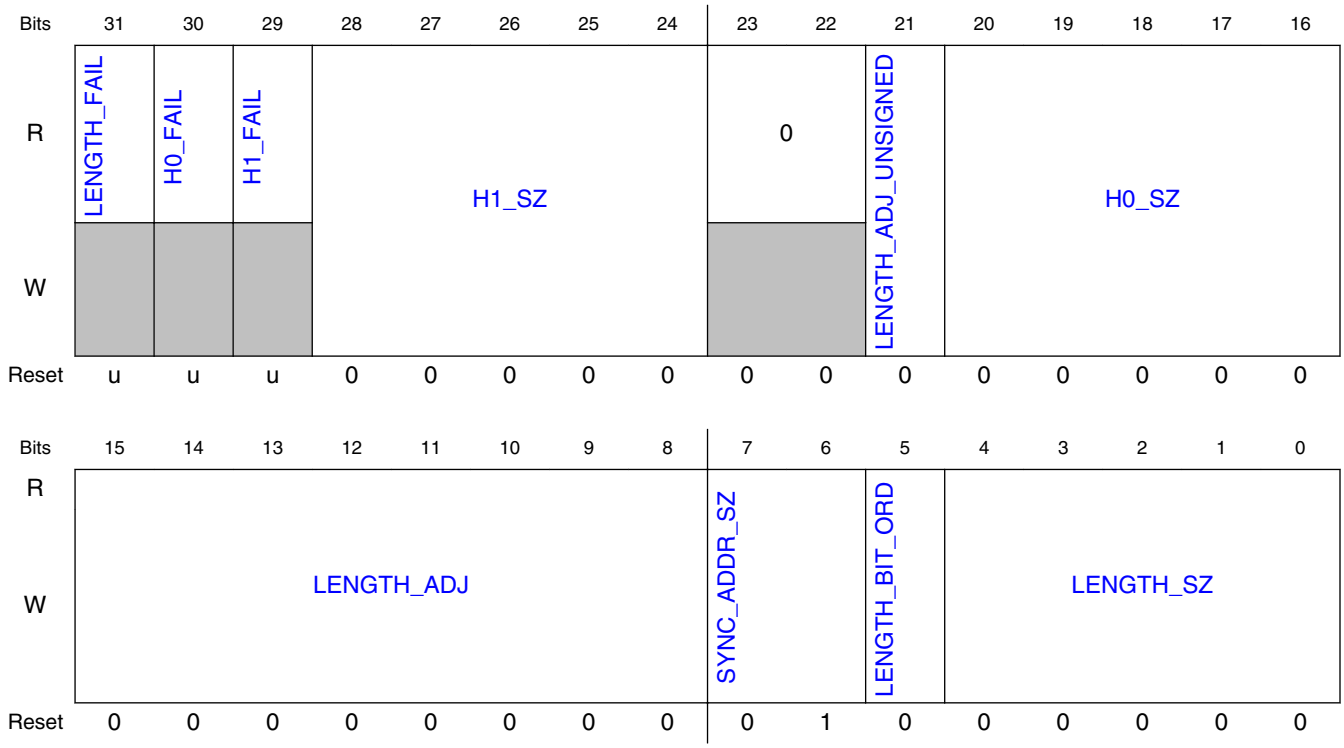
Field	Function
31-8 —	Reserved.
7-0 PART_ID	Part ID Part ID to identify HW revision of the Generic FSK Link Layer

45.5.2.2.1.20 PACKET CONFIGURATION (PACKET_CFG)

45.5.2.2.1.20.1 Offset

Register	Offset
PACKET_CFG	60h

45.5.2.2.1.20.2 Diagram



45.5.2.2.1.20.3 Fields

Field	Function
31 LENGTH_FAIL	Maximum Length Violated Status Bit For packets received with REC_BAD_PKT=1, LENGTH_FAIL indicates the extracted LENGTH header field exceeded LENGTH_MAX
30 H0_FAIL	H0 Violated Status Bit For packets received with REC_BAD_PKT=1, H0_FAIL indicates the received H0 header field violates the H0_MASK/H0_MATCH pattern
29 H1_FAIL	H1 Violated Status Bit For packets received with REC_BAD_PKT=1, H1_FAIL indicates the received H1 header field violates the H1_MASK/H1_MATCH pattern
28-24 H1_SZ	H1 Size Length of H1 in bits; 0 ≤ H1_SZ ≤ 16
23-22 —	Reserved.
21 LENGTH_ADJ_UNSIGNED	Length Adjustment Unsigned Enabled 0b - Hardware interprets LENGTH_ADJ as a signed integer (default) 1b - Hardware interprets LENGTH_ADJ as a unsigned integer
20-16 H0_SZ	H0 Size Size of H0 in bits; 0 ≤ H0_SZ ≤ 16

Table continues on the next page...

Field	Function
15-8 LENGTH_ADJ	Length Adjustment Signed Adjustment to the LENGTH field for TX and RX. A value of 0 (default) means LENGTH is interpreted as PAYLOAD + CRC
7-6 SYNC_ADDR_SZ	Sync Address Size Number of Octets = SYNC_ADDR_SZ + 1, 0 ≤ SYNC_ADDR_SZ ≤ 3.
5 LENGTH_BIT_ORD	LENGTH Bit Order Bit order for the LENGTH field of the header 0b - LS Bit First 1b - MS Bit First
4-0 LENGTH_SZ	LENGTH Size Size of LENGTH field of the header, in bits; 0 ≤ LENGTH_SZ ≤ 16

45.5.2.2.1.21 H0 CONFIGURATION (H0_CFG)

45.5.2.2.1.21.1 Offset

Register	Offset
H0_CFG	64h

45.5.2.2.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	H0_MASK															
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	H0_MATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.5.2.2.1.21.3 Fields

Field	Function
31-16 H0_MASK	H0 Mask Register For each bit that is set to 1, the received H0 field must match the corresponding bit of H0_MATCH[15:0], else the received packet is rejected.

Table continues on the next page...

Field	Function
15-0 H0_MATCH	H0 Match Register For each bit of H0_MASK[15:0] that is set to 1, the received H0 field must match this register, else the received packet is rejected.

45.5.2.2.1.22 H1 CONFIGURATION (H1_CFG)

45.5.2.2.1.22.1 Offset

Register	Offset
H1_CFG	68h

45.5.2.2.1.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	H1_MASK															
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	H1_MATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

45.5.2.2.1.22.3 Fields

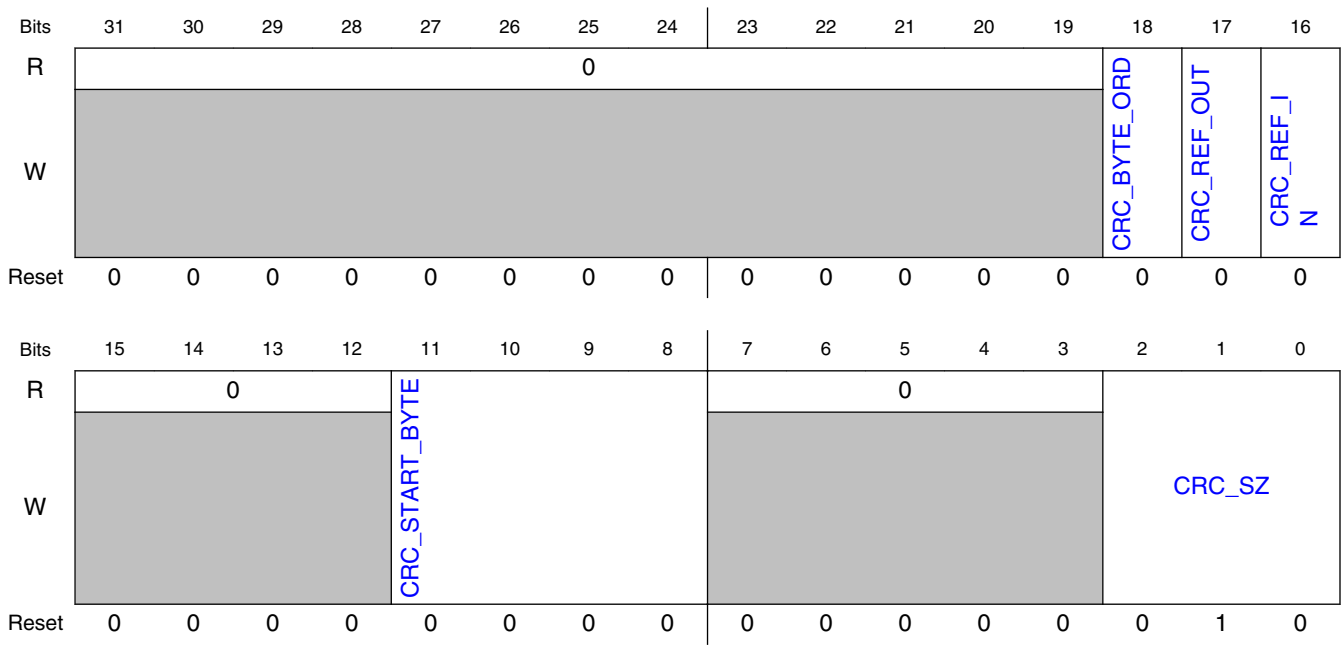
Field	Function
31-16 H1_MASK	H1 Mask Register For each bit that is set to 1, the received H1 field must match the corresponding bit of H1_MATCH[15:0], else the received packet is rejected.
15-0 H1_MATCH	H1 Match Register For each bit of H1_MASK[15:0] that is set to 1, the received H1 field must match this register, else the received packet is rejected.

45.5.2.2.1.23 CRC CONFIGURATION (CRC_CFG)

45.5.2.2.1.23.1 Offset

Register	Offset
CRC_CFG	6Ch

45.5.2.2.1.23.2 Diagram



45.5.2.2.1.23.3 Fields

Field	Function
31-19 —	Reserved.
18 CRC_BYTE_ORD	CRC Byte Order 0b - LS Byte First 1b - MS Byte First
17 CRC_REF_OUT	CRC Reflect Out 0b - do not manipulate CRC result 1b - CRC result is to be reflected bitwise (operated on entire word)
16 CRC_REF_IN	CRC Reflect In 0b - do not manipulate input data stream 1b - reflect each byte in the input stream bitwise
15-12 —	Reserved.
11-8	Configure CRC Start Point

Table continues on the next page...

Field	Function
CRC_START_BYTE	Start CRC with this byte position. Byte #0 is the first byte of Sync Address
7-3 —	Reserved.
2-0 CRC_SZ	CRC Size (in octets) Number of CRC Octets = CRC_SZ, 0 ≤ CRC_SZ ≤ 4.

45.5.2.2.1.24 CRC INITIALIZATION (CRC_INIT)

45.5.2.2.1.24.1 Offset

Register	Offset
CRC_INIT	70h

45.5.2.2.1.24.2 Diagram

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

45.5.2.2.1.24.3 Fields

Field	Function
31-0 CRC_SEED	CRC Seed Value Initial Value for CRC LFSR

45.5.2.2.1.25 CRC POLYNOMIAL (CRC_POLY)

45.5.2.2.1.25.1 Offset

Register	Offset
CRC_POLY	74h

45.5.2.2.1.25.2 Diagram

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

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

45.5.2.2.1.25.3 Fields

Field	Function
31-0 CRC_POLY	CRC Polynomial.

45.5.2.2.1.26 CRC XOR OUT (CRC_XOR_OUT)

45.5.2.2.1.26.1 Offset

Register	Offset
CRC_XOR_OUT	78h

45.5.2.2.1.26.2 Diagram

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

45.5.2.2.1.26.3 Fields

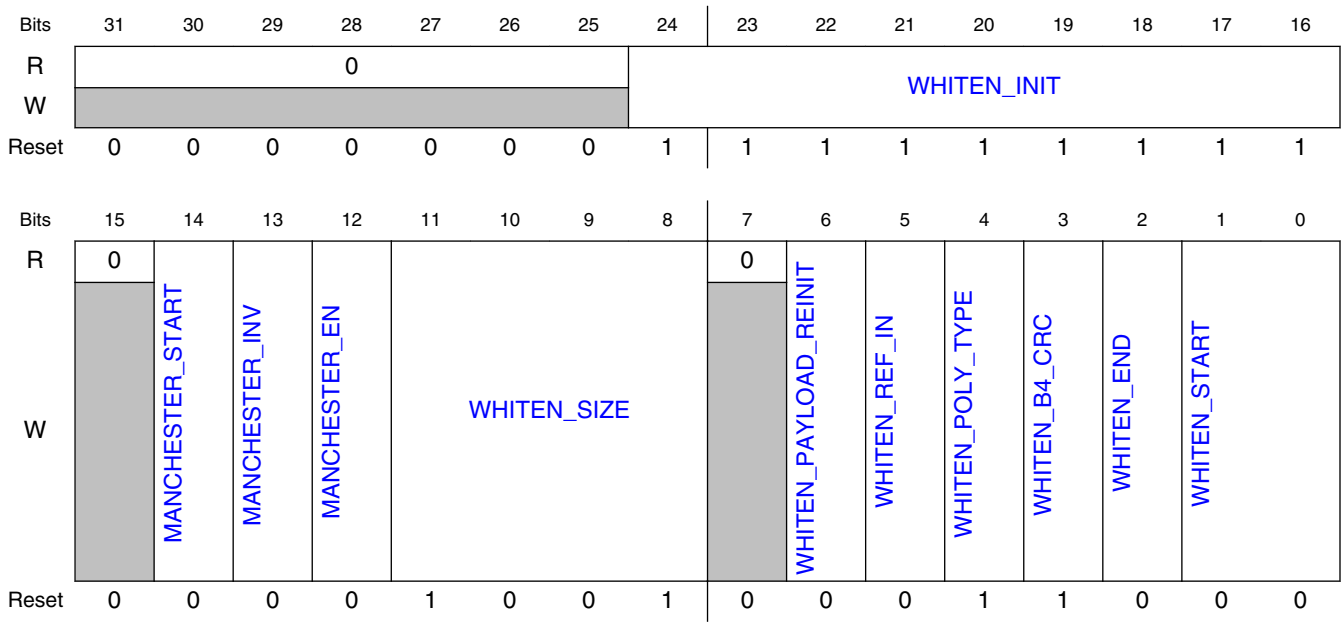
Field	Function
31-0 CRC_XOR_OUT	CRC XOR OUT Register XOR mask for CRC result (for no mask, should be 0)

45.5.2.2.1.27 WHITENER CONFIGURATION (WHITEN_CFG)

45.5.2.2.1.27.1 Offset

Register	Offset
WHITEN_CFG	7Ch

45.5.2.2.1.27.2 Diagram



45.5.2.2.1.27.3 Fields

Field	Function
31-25 —	Reserved.
24-16 WHITEN_INIT	Initialization Value for Whitening/De-whitening
15 —	Reserved.
14 MANCHESTER_START	Configure Manchester Encoding Start Point 0b - Start Manchester coding at start-of-payload 1b - Start Manchester coding at start-of-header
13 MANCHESTER_INV	Configure for Inverted Manchester Encoding 0b - Manchester coding as per 802.3 1b - Manchester coding as per 802.3 but with the encoding signal inverted
12 MANCHESTER_EN	Configure for Manchester Encoding/Decoding 0b - Disable Manchester encoding (TX) and decoding (RX) 1b - Enable Manchester encoding (TX) and decoding (RX)
11-8 WHITEN_SIZE	Length of Whitener LFSR
7 —	Reserved.
6	Configure for Whitener re-initialization 0b - Don't re-initialize Whitener LFSR at start-of-payload

Table continues on the next page...

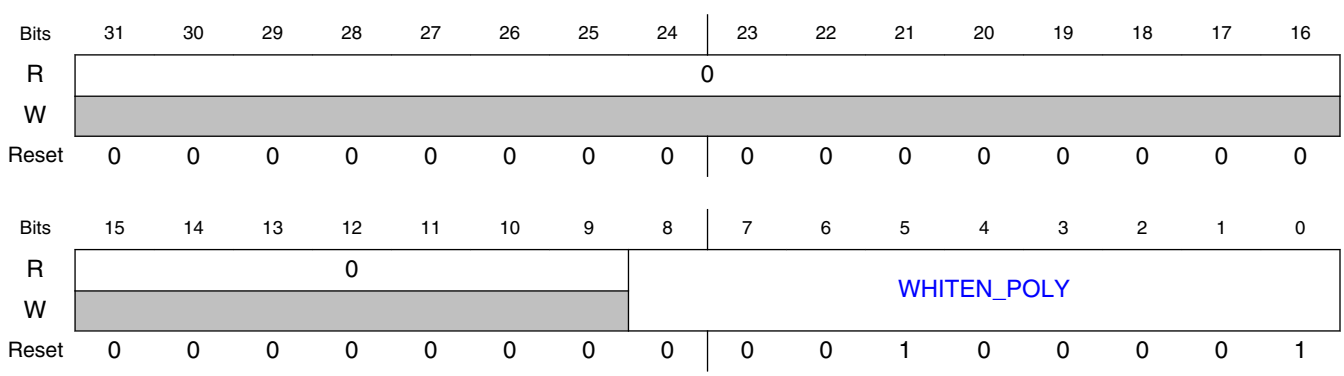
Field	Function
WHITEN_PAYLOAD_REINIT	1b - Re-initialize Whitener LFSR at start-of-payload
5 WHITEN_REF_IN	Whiten Reflect Input The input data stream is reflected, bit-wise, per byte, if this register bit is asserted. Bit 7 becomes bit 0, bit 6 becomes bit 1, etc. This register bit will only cause the reflection of the payload data bits as they are used in the whiten calculation and will not cause any change in the output bit order.
4 WHITEN_POLY_TYPE	Whiten Polynomial Type A Fibonacci type LFSR is used with the whiten polynomial if this register bit is asserted. Otherwise, a Galois type LFSR is used.
3 WHITEN_B4_CRC	Congifure for Whitening-before-CRC Sets the order of Bit Stream Processing for TX and RX. 0b - CRC before whiten/de-whiten 1b - Whiten/de-whiten before CRC
2 WHITEN_END	Configure end-of-whitening 0b - end whiten at end-of-payload 1b - end whiten at end-of-crc
1-0 WHITEN_START	Configure Whitener Start Point 00b - no whitening 01b - start whitening at start-of-H0 10b - start whitening at start-of-H1 but only if LENGTH > WHITEN_SZ_THR 11b - start whitening at start-of-payload but only if LENGTH > WHITEN_SZ_THR

45.5.2.2.1.28 WHITENER POLYNOMIAL (WHITEN_POLY)

45.5.2.2.1.28.1 Offset

Register	Offset
WHITEN_POLY	80h

45.5.2.2.1.28.2 Diagram



45.5.2.2.1.28.3 Fields

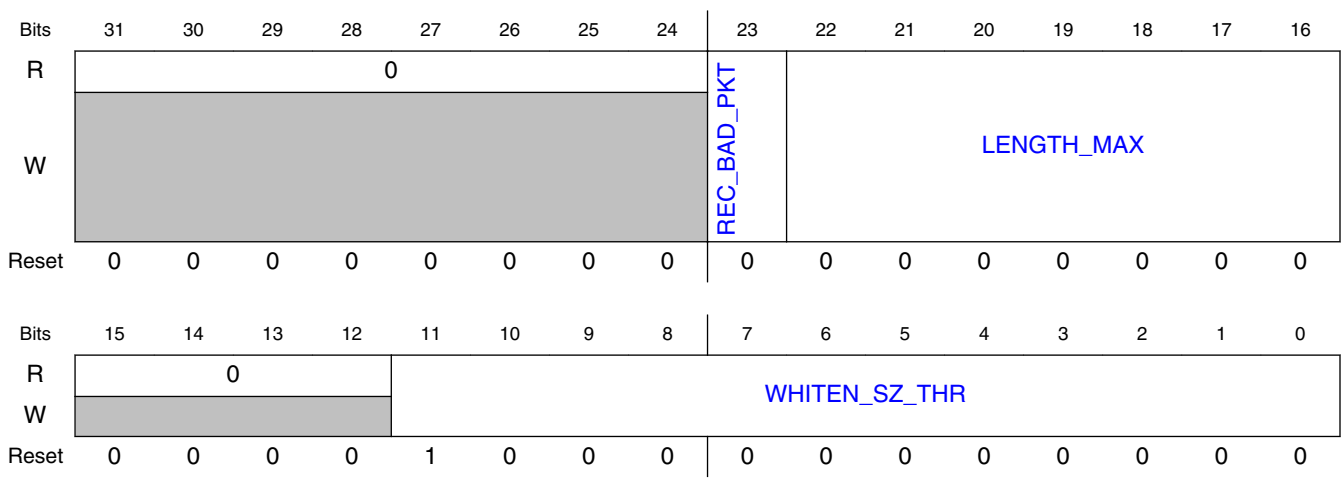
Field	Function
31-9 —	Reserved.
8-0 WHITEN_POLY	Whitener Polynomial The 9-bit polynomial used in the whiten calculation. The polynomial value must be right-justified if smaller than 9-bits.

45.5.2.2.1.29 WHITENER SIZE THRESHOLD (WHITEN_SZ_THR)

45.5.2.2.1.29.1 Offset

Register	Offset
WHITEN_SZ_THR	84h

45.5.2.2.1.29.2 Diagram



45.5.2.2.1.29.3 Fields

Field	Function
31-24 —	Reserved.
23 REC_BAD_PKT	Receive Bad Packets Enable Packets which fail H0-filtering, H1-filtering, or Maximum Length-filtering, to be fully received without an RX recycle (intended for debug purposes)

Table continues on the next page...

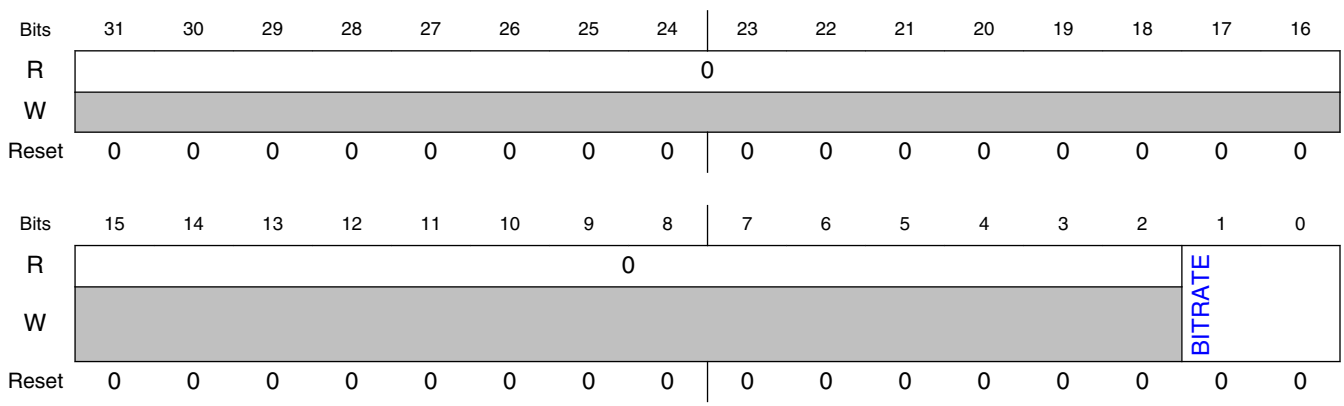
Field	Function
	0b - packets which fail H0, H1, or LENGTH_MAX result in an automatic recycle after the header is received and parsed 1b - packets which fail H0, H1, or LENGTH_MAX are received in their entirety
22-16 LENGTH_MAX	Maximum Length for Received Packets Sets the Maximum Length Packet that can be received, in multiples of 16 bytes. LENGTH_MAX is compared directly against the extracted LENGTH field of the header (not the adjusted length). LENGTH_MAX=0 (default) is a special case that implies no limit.
15-12 —	Reserved.
11-0 WHITEN_SZ_THR	Whitener Size Threshold Minimum Packet Length (extracted LENGTH field) required to enable whiten. Requires WHITEN_START=2 or 3

45.5.2.2.1.30 BIT RATE (BITRATE)

45.5.2.2.1.30.1 Offset

Register	Offset
BITRATE	88h

45.5.2.2.1.30.2 Diagram



45.5.2.2.1.30.3 Fields

Field	Function
31-2 —	Reserved.

Table continues on the next page...

Field	Function
1-0 BITRATE	Bit Rate Selects the Bit Rate for the Generic FSK Link Layer. The blocks which interface to the Link Layer (e.g., PHY, CRC/Whitener), also adapt to the selected bit rate. 00b - 1Mbit/sec 01b - 500Kbit/sec 10b - 250Kbit/sec (not supported if WHITEN_CFG[MANCHESTER_EN]=1)

45.5.2.2.1.31 PACKET BUFFER PARTITION POINT (PB_PARTITION)

45.5.2.2.1.31.1 Offset

Register	Offset
PB_PARTITION	8Ch

45.5.2.2.1.31.2 Diagram

Bits	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

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

45.5.2.2.1.31.3 Fields

Field	Function
31-11 —	Reserved.
10-0 PB_PARTITION	Packet Buffer Partition Point The Packet Buffer Partition Point defines the starting point for the RX segment of the Packet Buffer. The partitioning of the consolidated RAM between TX and RX is controlled by this configurable TX/RX Partition Point. PB_PARTITION can be programmed with any value between 0 (base of RAM0) and 1088 (after the last address of RAM1). The consolidated RAM is partitioned such that the TX buffer resides before the Partition Point (RAM entries 0 to (PB_PARTITION-1)), and the RX buffer resides after it (RAM entries PB_PARTITION to 1087). Programming the Partition Point register to 0 dedicates the entire consolidated RAM to RX; programming the Partition Point register to 1088 dedicates the entire consolidated RAM to TX; programming the Partition Point register to any value between 1 and 1087

Field	Function
	yields a split TX/RX buffer. The TX segment (if any) is always first in the Packet Buffer, starting at Packet RAM address 0 of RAM0. Since the Packet Buffer RAM word-width is 2 bytes, the units for PB_PARTITION is "words". (Multiply by 2 to convert to byte addressing)

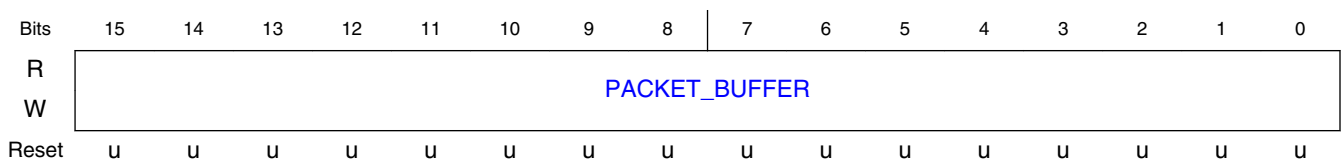
45.5.2.2.1.32 PACKET BUFFER (PACKET_BUFFER_0 - PACKET_BUFFER_1087)

45.5.2.2.1.32.1 Offset

For a = 0 to 1087:

Register	Offset
PACKET_BUFFER_a	700h + (a × 2h)

45.5.2.2.1.32.2 Diagram



45.5.2.2.1.32.3 Fields

Field	Function
15-0 PACKET_BUFFER	PACKET BUFFER RAM Storage for packet data.

45.5.2.2.2 Register Summary

45.5.2.2.3 Register Descriptions

Field	R/W	Description
SEQCMD[3:0]	w	Sequence Command Register. See Section Sequence Commands and Status for details

Table continues on the next page...

Field	R/W	Description
SEQSTS[12:0]	r	Sequence Status Register. See Section Sequence Commands and Status for details
CMDDEC_CS[2:0]	r	Current State of the Command Decoder FSM (debug only)
TX_WHITEN_DIS	rw	Disable all whitening on TX packets
RX_DEWHITEN_DIS	rw	Disable all de-whitening on RX packets
SW_CRC_EN	rw	SW override of the HW-computed CRC for TX. SW must include CRC in the Packet Buffer TX buffer
CRC_VALID	r	CRC Valid indicator for RX packets. 1: good CRC 0: bad CRC
TX_WARMUP[7:0]	r	TX warmup time, in microseconds, provided to Link Layer SW
RX_WARMUP[7:0]	r	RX warmup time, in microseconds, provided to Link Layer SW
CHANNEL_NUM[6:0]	rw	RF Channel Select: 0 <= CHANNEL_NUM <= 127; Formula: F = (2360 + CHANNEL_NUM) [in MHz]
TX_POWER[5:0]	rw	PA Power Level. Refer Data Sheet for mapping to dBm.
RSSI[7:0]	r	Received Signal Stength Indicator, signed value
LQI[7:0]	r	Link Quality Indicator, signed value
PREAMBLE_SZ[2:0]	rw	Number of Octets = PREAMBLE_SZ + 1, where 0 <= PREAMBLE_SZ <= 7
BYTE_COUNTER[12:0]	r	Signed value, reflects the current Byte Count, for TX and RX. < 0: Preamble byte(s), TX only 0: First Byte of Network Address; 1: Second Byte of Network Address …
XCVR_BUSY	r	For multi-protocol arbitration, XCVR_BUSY=1 indicates an RF channel access is underway (TSM is busy)
EVENT_TMR[23:0]	r +load + add	Event Timer can be read in these byte locations. To update the Event Timer, either: 1. Write the desired EVENT_TMR to these bytes and set EVENT_TMR_LD=1, or, 2. Write the desired EVENT_TMR *increment* amount to these bytes and set EVENT_TMR_ADD=1 Note for EVENT_TMR_ADD, EVENT_TMR[23:0] is a signed value.
EVENT_TMR_LD	w	A write access with this bit sets loads EVENT_TMR with the contents of EVENT_TMR[23:0]
EVENT_TMR_ADD	w	A write access with this bit sets increments EVENT_TMR by the contents of EVENT_TMR[23:0]. This is a signed addition.
TIMESTAMP[23:0]	r	Received Packet Timestamp, Captured from EVENT_TMR at NTW_ADR_IRQ.
T1_CMP[23:0]	rw	Timer1 (T1) Compare Value. Can be used to generate T1_IRQ and/or launch Sequence Commands
T2_CMP[23:0]	rw	Timer2 (T2) Compare Value. Can be used to generate T2_IRQ and/or launch Sequence Commands

Table continues on the next page...

Field	R/W	Description
T1_CMP_EN	rw	Enable Timer Compare #1 (T1_CMP) to generate T1_IRQ and/or launch Sequence Commands
T2_CMP_EN	rw	Enable Timer Compare #2 (T2_CMP) to generate T2_IRQ and/or launch Sequence Commands
NTW_ADR_MCH[3:0]	r	Indicates which of the 4 Network Addresses has matched in the PHY. Valid during a RX sequence at the point of match, and remains valid until either: 1. The next RX sequence begins (if the current packet passed CRC and header filtering), or, 2. An RX recycle to Network Address search (if the current packet failed CRC or header filtering)
NTW_ADR0[31:0]	rw	Network Address 0, to PHY for correlation
NTW_ADR1[31:0]	rw	Network Address 1, to PHY for correlation
NTW_ADR2[31:0]	rw	Network Address 2, to PHY for correlation
NTW_ADR3[31:0]	rw	Network Address 3, to PHY for correlation
NTW_ADR_EN[3:0]	rw	Enable Network Address N for PHY correlation, where 0 <= N <= 3. Any bit combination can be set.
NTW_ADR0_SZ[1:0]	rw	3: Network Address 0 requires a 32-bit correlation 2: Network Address 0 requires a 24-bit correlation 1: Network Address 0 requires a 16-bit correlation 0: Network Address 0 requires a 8-bit correlation
NTW_ADR1_SZ[1:0]	rw	3: Network Address 1 requires a 32-bit correlation 2: Network Address 1 requires a 24-bit correlation 1: Network Address 1 requires a 16-bit correlation 0: Network Address 1 requires a 8-bit correlation
NTW_ADR2_SZ[1:0]	rw	3: Network Address 2 requires a 32-bit correlation 2: Network Address 2 requires a 24-bit correlation 1: Network Address 2 requires a 16-bit correlation 0: Network Address 2 requires a 8-bit correlation
NTW_ADR3_SZ[1:0]	rw	3: Network Address 3 requires a 32-bit correlation 2: Network Address 3 requires a 24-bit correlation 1: Network Address 3 requires a 16-bit correlation 0: Network Address 3 requires a 8-bit correlation
NTW_ADR_THR0[2:0]	rw	Number of Tolerated bit errors for Network Address 0
NTW_ADR_THR1[2:0]	rw	Number of Tolerated bit errors for Network Address 1
NTW_ADR_THR2[2:0]	rw	Number of Tolerated bit errors for Network Address 2
NTW_ADR_THR3[2:0]	rw	Number of Tolerated bit errors for Network Address 3
RX_WATERMARK_IRQ	r / w1tc	Asserts when Byte Counter == RX_WATERMARK[12:0]
NTW_ADR_IRQ	r / w1tc	Network Address Match Interrupt. A NA Match has occurred
TX_IRQ	r / w1tc	TX Interrupt. The TX sequence has completed with a successful packet transmission.
RX_IRQ	r / w1tc	RX Interrupt. The RX sequence has completed with a successful packet reception.

Table continues on the next page...

Field	R/W	Description
PLL_UNLOCK_IRQ	r / w1tc	PLL Unlock Interrupt. An unlock event has occurred.
SEQ_END_IRQ	r / w1tc	Sequence End Interrupt. Will assert when any TX or RX sequence ends for any reason.
T1_IRQ	r / w1tc	Timer Compare #1 (T1) Interrupt
T2_IRQ	r / w1tc	Timer Compare #2 (T2) Interrupt
WAKE_IRQ	r / w1tc	Wake Interrupt. A WAKE_IRQ will be triggered when the GENERIC_FSK Link Layer Controller has awoken from a DSM (Deep Sleep Mode) cycle. WAKE_IRQ indicates that the RF Oscillator has been restarted, and the GENERIC_FSK EVENT_TMR has resumed counting.
TSM_IRQ	r	TSM Interrupt. Indicates TSM0_IRQ or TSM1_IRQ is set in XCVR_STATUS. For debug purposes.
RX_WATERMARK_IRQ_EN	rw	Enable for RX_WATERMARK_IRQ
NTW_ADR_IRQ_EN	rw	Enable for NTW_ADR_IRQ
TX_IRQ_EN	rw	Enable for TX_IRQ
RX_IRQ_EN	rw	Enable for RX_IRQ
PLL_UNLOCK_IRQ_EN	rw	Enable for PLL_UNLOCK_IRQ
SEQ_END_IRQ_EN	rw	Enable for SEQ_END_IRQ
T1_IRQ_EN	rw	Enable for T1_IRQ
T2_IRQ_EN	rw	Enable for T2_IRQ
WAKE_IRQ_EN	rw	Enable for WAKE_IRQ
TSM_IRQ_EN	rw	Enable for TSM_IRQ
GENERIC_FSK_IRQ_EN	rw	Master enable for the GENERIC_FSK_IRQ interrupt line to the MCU.
CRC_IGNORE	rw	If set, assert RX_IRQ even for a received packet which fails CRC verification, and do not recycle.
RX_WATERMARK[12:0]	rw	Sets the trigger for RX_WATERMARK_IRQ. Trigger the RX_WATERMARK_IRQ when: RX Byte Counter == RX_WATERMARK[12:0]
GEN_SLEEP_REQUEST	w	Enable a match on MAN_SLEEP[23:0] to DSM_TIMER[23:0], to enter Deep Sleep Mode
DSM_TIMER[23:0]	r	Current State of the 32 KHz Sleep Timer **NOTE:** This register resides in RSIM Address Space
PART_ID[7:0]	r	Part ID to identify HW revision
SYNC_ADDR_SZ[1:0]	rw	Number of Octets = SYNC_ADDR_SZ + 1, 0 <= SYNC_ADDR_SZ <= 3. Applies to TX only.
H0_SZ[4:0]	rw	Size of H0 in bits; 0 <= H0_SZ <= 16
H0_MASK[15:0]	rw	For each bit that is set to 1, the received H0 field must match the corresponding bit of H0_MATCH[15:0], else the received packet is rejected.
H0_MATCH[15:0]	rw	For each bit of H0_MASK[15:0] that is set to 1, the received H0 field must match this register, else the received packet is rejected.
LENGTH_SZ[4:0]	rw	Size of Packet Length in bits; 0 <= LENGTH_SZ <= 16
LENGTH_BIT_ORD	rw	Bit order for the LENGTH field 0: LS Bit First;

Table continues on the next page...

Field	R/W	Description
		1: MS Bit First
LENGTH_ADJ[7:0]	rw	Signed Adjustment to the LENGTH field for TX and RX. A value of 0 (default) means LENGTH is interpreted as PAYLOAD + CRC
LENGTH_ADJ_UNSIGNED	rw	Length Adjustment Unsigned Enabled 0: Hardware interprets LENGTH_ADJ as a signed integer (default) 1: Hardware interprets LENGTH_ADJ as a unsigned integer
LENGTH_MAX[6:0]	rw	Sets the Maximum Length Packet that can be received, in multiples of 16 bytes. LENGTH_MAX is compared directly against the extracted LENGTH field of the header (not the adjusted length). LENGTH_MAX=0 (default) is a special case that implies no limit.
REC_BAD_PKT	rw	1: packets which fail H0, H1, or LENGTH_MAX are received in their entirety 0: packets which fail H0, H1, or LENGTH_MAX result in a recycle after the header is fully received
LENGTH_FAIL	r	For packets received with REC_BAD_PKT=1, LENGTH_FAIL indicates the extracted LENGTH header field exceeded LENGTH_MAX
H0_FAIL	r	For packets received with REC_BAD_PKT=1, H0_FAIL indicates the received H0 header field violates the H0_MASK/H0_MATCH pattern
H1_FAIL	r	For packets received with REC_BAD_PKT=1, H1_FAIL indicates the received H1 header field violates the H1_MASK/H1_MATCH pattern
H1_SZ[4:0]	rw	Length of H1 in bits; 0 <= H1_SZ <= 16
H1_MASK[15:0]	rw	For each bit that is set to 1, the received H1 field must match the corresponding bit of H1_MATCH[15:0], else the received packet is rejected.
H1_MATCH[15:0]	rw	For each bit of H1_MASK[15:0] that is set to 1, the received H1 field must match this register, else the received packet is rejected.
CRC_SZ[2:0]	rw	Number of CRC Octets = CRC_SZ, 0 <= CRC_SZ <= 4.
CRC_POLY[31:0]	rw	CRC Polynomial. 1: XOR exists in the bit's feedback path; 0: no XOR in the bit's feedback path
CRC_SEED[31:0]	rw	CRC Seed Value
CRC_START_BYTE[3:0]	rw	Start CRC with this byte position. Byte #0 is the first byte of Sync Address
CRC_BYTE_ORD	rw	0: LS Byte First 1: MS Byte First
CRC_REF_IN	rw	0: do not manipulate input data stream; 1: reflect each byte in the input stream bitwise
CRC_REF_OUT	rw	0: do not manipulate CRC result; 1: CRC result is to be reflected bitwise (operated on entire word)
CRC_XOR_OUT[31:0]	rw	XOR mask for CRC result (for no mask, should be 0)
WHITEN_B4_CRC	rw	Order of Bit Stream Processing for TX and RX: 1: Whiten/de-whiten before CRC 0: CRC before whiten/de-whiten
WHITEN_SEED[8:0]	rw	Initialization Value for Whitening/De-whitening

Table continues on the next page...

Field	R/W	Description
WHITEN_ POLY[8:0]	rw	Polynomial Value for Whitening/De-whitening
WHITEN_POLY_TYPE	rw	Whiten polynomial type. A Fibonacci type LFSR is used with the whiten polynomial if this bit is asserted. Otherwise, a Galois type LFSR is used.
WHITEN_SIZE[3:0]	rw	Whitener Length
WHITEN_REF_IN	rw	The input data stream is reflected, bit-wise, per byte, if this bit is asserted. Bit 7 becomes bit 0, bit 6 becomes bit 1, etc. This bit will only cause the reflection of the payload data bits as they are used in the whiten calculation and will not cause any change in the output bit order
WHITEN_START[1:0]	rw	0: no whiten; 1: start whiten @ start-of-H0; 2: start whiten @ start-of-H1 but only if LENGTH > WHITEN_SZ_THR 3: start whiten @ start-of-payload but only if LENGTH > WHITEN_SZ_THR
WHITEN_END	rw	0: end whiten at end-of-payload 1: end whiten at end-of-crc
WHITEN_PAYLOAD_REINIT	rw	1: re-assert whiten_init at start-of-payload 0 = don't re-assert whiten_init
WHITEN_SZ_THR[11:0]	rw	Minimum Packet Length (extracted LENGTH field) required to enable whiten. Requires WHITEN_START=2 or 3
BITRATE[1:0]	rw	0: 1 Mbs 1: 500 Kbs 2: 250 Kbs
MANCHESTER_EN	rw	1: Enable Manchester encoding (TX) and decoding (RX) 0: Disable Manchester Encoding and Decoding
MANCHESTER_INV	rw	0: Manchester coding as per 802.3 1: Manchester coding as per 802.3 but with the encoding signal inverted
MANCHESTER_START	rw	0: Start Manchester coding @ start-of-payload 1: Start Manchester coding @ start-of-header

45.5.2.3 Functional Description

45.5.2.3.1 Packet Configuration

The Generic FSK Link Layer Controller provides the capability to configure the over-the-air packet structure by way of a set of programmable registers. The Generic FSK packet has a consistent structure, which consists of various elements which are transmitted and received in fixed order. Most of the packet elements have associated configurability,

which allows software to pre-configure, for example, the size of the element (in number of bits, or octets), or bit ordering of the element. Other configuration options apply to the entire packet, such as bitrate. In addition, the generic structure defines a variable-length header, for which some (optional) primitive parsing is provided, allowing hardware to differentiate packets based on header-bit settings, into compliant and non-compliant packets. Data Indication is provided on compliant packets only. Non-complaint packets are discarded. For transmitting and receiving, multiple bitrate options are provided. By default, all packet elements are transmitted and received LSB-first.

The Generic FSK Link Layer Controller supports a packet structure based on the following template:

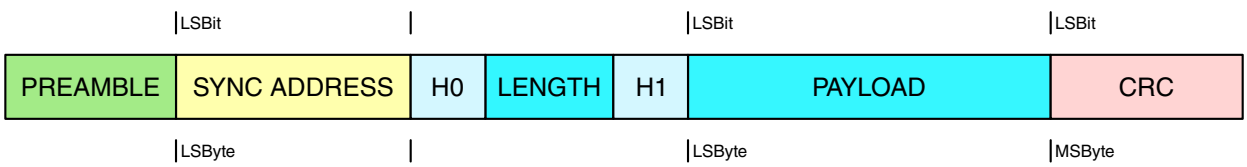


Figure 45-101. GENERIC_FSK Packet Structure

The Generic_FSK Link Layer controller transmits and receives packets which conform to the format shown above. The elements of the packet are transmitted in received in the order shown, although some of the packet elements may be optionally skipped, depending on the configuration options chosen.

The following diagram depicts the register configurability associated with the Preamble, Sync Address, H0, Length, and H1 elements of the packet structure.

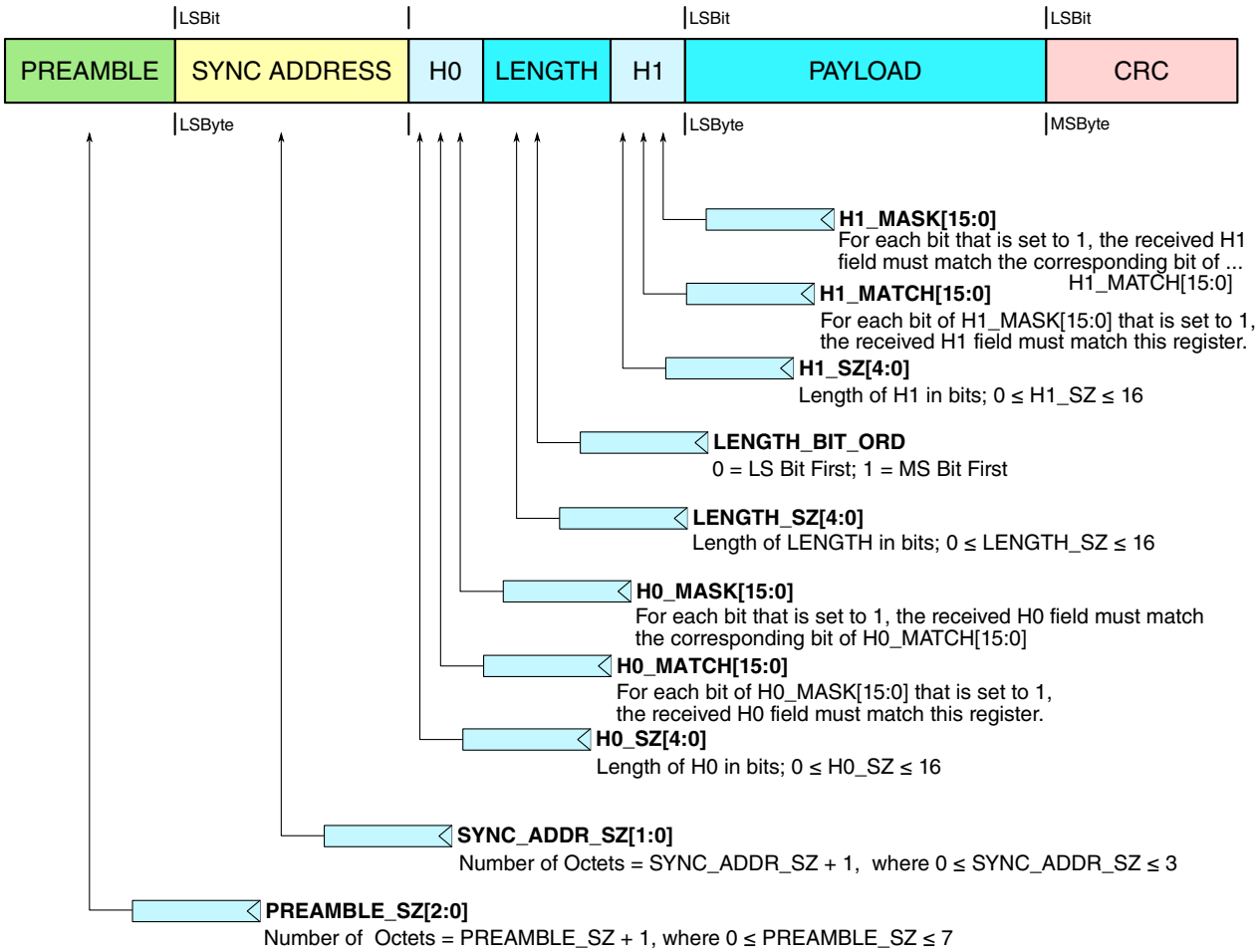


Figure 45-102. Packet Structure Definitions (Part 1)

The following diagram depicts the register configurability associated with the Payload and CRC elements of the packet structure.

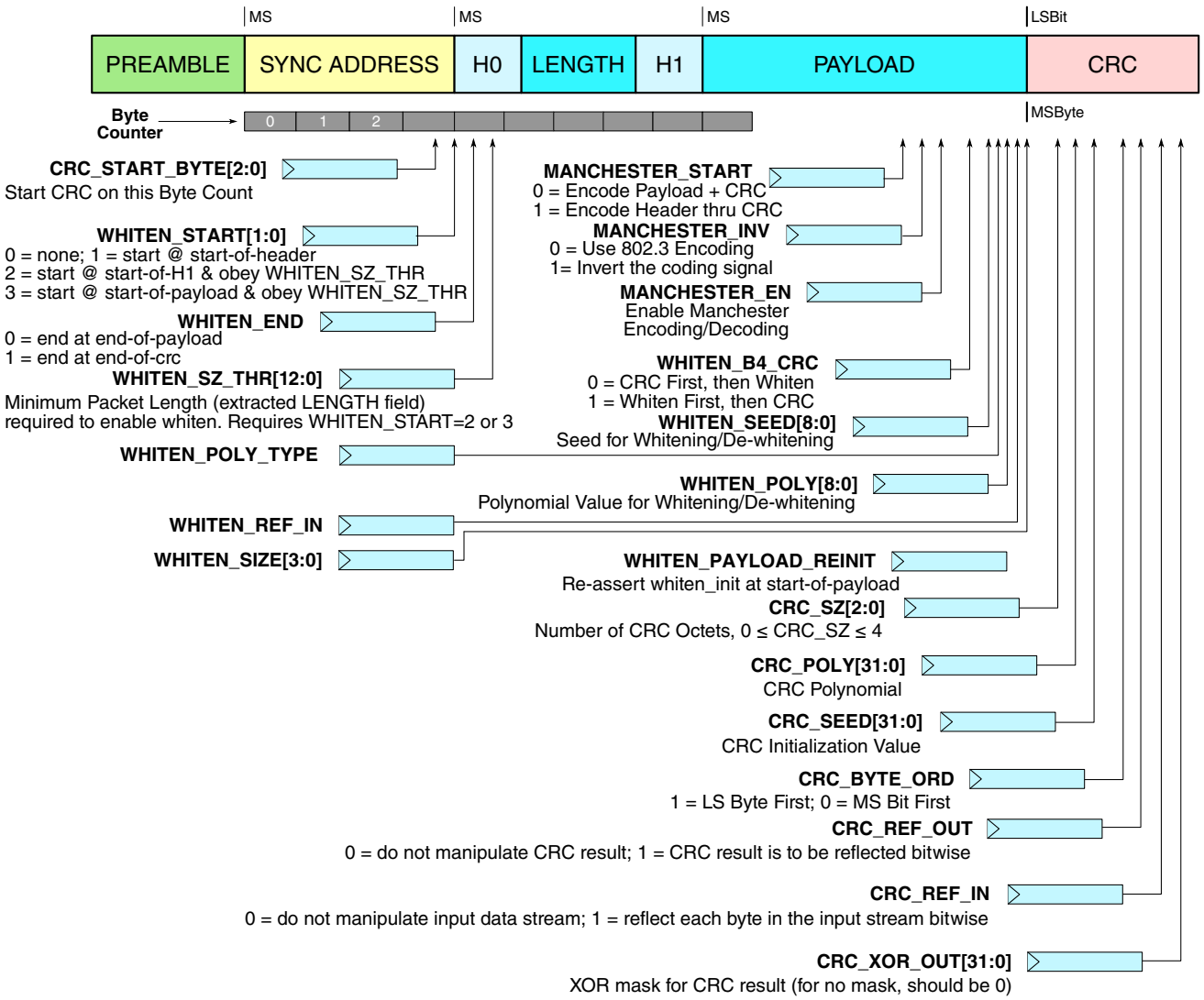


Figure 45-103. Packet Structure Definitions (Part 2)

The following subsections describe each of the packet elements, and the configurability associated with each.

PREAMBLE

The preamble pattern is 0x55 or 0xAA. Like all packet elements, preamble is transmitted LSB first. The Sync Address, which immediately follows the preamble, determines which of the 2 preamble options is selected by hardware: the controller hardware selects the preamble pattern based on the first transmitted bit of Network Address, such that the last bit of preamble is the opposite polarity from the first bit of Network Address, forcing a bit transition at this boundary. The number of octets of preamble is programmable, via the PREAMBLE_SZ[2:0] register, with a range from 1 to 8 octets. A setting of PREAMBLE_SZ=0 yields a single preamble octet, whereas a setting of PREAMBLE_SZ=7 yields 8 octets. For transmission, the Generic FSK

BYTE_COUNTER will be negative during preamble transmission. If PREAMBLE_SZ=0 (1 preamble octet), BYTE_COUNTER will be -1 while the single preamble octet is being transmitted. If PREAMBLE_SZ=7 (8 octets), BYTE_COUNTER will be -8 during the first preamble octet, -7 during the next, and so on and so forth, until the eighth and final octet, during which the BYTE_COUNTER will be -1. Unlike most packet elements, the preamble is generated and shifted out by hardware (only preamble and CRC are hardware-generated); there is no software setup required, and preamble does not appear in the Packet Buffer TX buffer. During reception, the Link Layer controller does not receive preamble, since preamble detection is performed by the PHY; preamble does not appear in the Packet Buffer RX buffer.

SYNC (NETWORK) ADDRESS

Sync Address, known synonymously as Network Address or Access Address depending on the protocol in use, is the second packet element transmitted by the Link Layer controller, and the first element received. In a network of associated devices, all devices generally share a common Network Address. The first bit of Network Address determines the preamble selection for transmission, as noted above. For TX, Network Address is the first packet element programmed into the Packet Buffer TX buffer, in LSB-first format. For RX, Network Address is also the first element stored into the Packet Buffer RX buffer, in LSB-first format. The register NTW_ADRx_SZ[1:0] determines the length, in octets, of the Network Address field in the packet structure. A setting of NTW_ADRx_SZ=0 yields a single-octet Network Address, whereas a setting of NTW_ADRx_SZ=3 yields 4 octets. For reception, up to four unique Network Addresses are supported. Any combination of the 4 can be simultaneously searched for. Network Address correlation is performed in the PHY. Multiple Network Address capability allows GENERIC_FSK devices to be associated to more than one network simultaneously, if the network topology supports it; this feature also enables multi-protocol operation. Each of the 4 Network Address options has its own enable bit, NTW_ADR_EN[x]; when the enable bit for a Network Address is set to 1, the PHY is instructed to search for that Network Address pattern in the demodulated data, and when the enable bit is set to 0, the PHY will disregard that pattern. Each of the 4 Network Address options has its own NTW_ADRx_SZ register, to program its octet size during reception. Additionally, each of the 4 Network Address options has a register, NTW_ADR_THRx[2:0], which determine the number of bit errors that can be tolerated by the PHY during Network Address correlation to the respective Network Address option. Finally, when the PHY detects a match to one of the 4 Network Address options, with a number of bit errors less than or equal to the threshold programmed into NTW_ADR_THRx, a read only status bit, NTW_ADR_MCH[x] becomes set, indicating which of the 4 patterns matched. After a match has been detected, the PHY shall hold NTW_ADR_MCH[x] valid (sticky), until the next receiver warmup, or the next RX recycle. (Specifically, the TSM output rx_init clears the NTW_ADR_MCH[x] bits; TSM

will assert this signal during either an RX warmup or a recycle). A table listing the registers which apply to Network Address Management, for each of the 4 Network Address options, is shown below.

NTW_ADR PATTERN REGISTER	NTW_ADR ENABLE BIT	NTW_ADR SIZE REGISTER	NTW_ADR THRESHOLD REGISTER	NTW_ADR PATTERN-MATCH STATUS BIT
NTW_ADR_0[31:0]	NTW_ADR_EN[0]	NTW_ADR0_SZ[1:0]	NTW_ADR_THR0[2:0]	NTW_ADR_MCH[0]
NTW_ADR_1[31:0]	NTW_ADR_EN[1]	NTW_ADR0_SZ[1:0]	NTW_ADR_THR1[2:0]	NTW_ADR_MCH[1]
NTW_ADR_2[31:0]	NTW_ADR_EN[2]	NTW_ADR2_SZ[1:0]	NTW_ADR_THR2[2:0]	NTW_ADR_MCH[2]
NTW_ADR_3[31:0]	NTW_ADR_EN[3]	NTW_ADR3_SZ[1:0]	NTW_ADR_THR3[2:0]	NTW_ADR_MCH[3]

For Network Addresses of size less than 4 octets, the PHY expects the desired pattern to occupy the least-significant byte positions of the NTW_ADR_x register. Unused byte positions may be filled with any value.

NOTE

When programming the NTW_ADR_x, NTW_ADR_EN[x], NTW_ADRx_SZ, and NTW_ADR_THRx for any RX operation, the following software restrictions apply:

1. All NTW_ADR_x patterns which are enabled by their respective NTW_ADR_EN[x]=1 must be unique (disabled NTW_ADR_x registers need not follow this restriction).
2. Given restriction #1 above, not only must the patterns be unique, but the bit patterns of NTW_ADR_x and NTW_ADR_y must differ by an amount greater than the sum of the bit-error thresholds for each pattern, i.e., NTW_ADR_THRx + NTW_ADR_THRy.
3. Given all combinations of NTW_ADR_x and NTW_ADR_y, where (x != y) and both Network Addresses are enabled by NTW_ADR_EN[x]=NTW_ADR_EN[y]=1, if the sizes of the Network Address patterns differ because NTW_ADRx_SZ != NTW_ADRy_SZ, then the “matching portion” of NTW_ADR_x and NTW_ADR_y, that is, the octets of the pattern which the PHY is instructed to correlate to for each pattern, programmed into the lower byte positions of NTW_ADR_x or NTW_ADR_y for patterns of size less than 4 octets, must be unique.
4. Given restriction #3 above, not only must the “matching portion” of the patterns be unique, but bit patterns of the matching portions of NTW_ADR_x and NTW_ADR_y must differ by greater than the sum of the bit-error thresholds for each pattern, i.e., NTW_ADR_THRx + NTW_ADR_THRy.

These restrictions are designed to preclude multiple, simultaneous pattern matches during a pattern search. Results are indeterminate if the restrictions are violated.

When the PHY detects a pattern match on any enabled NTW_ADR_x during reception, and asserts NTW_ADR_MCH[x], the GENERIC_FSK Link Layer Controller shall assert NTW_ADR_IRQ. The NTW_ADR_IRQ is the Link Layer's first indication that a packet is being received. After any NTW_ADR_IRQ assertion, a packet will be received and stored into the Packet Buffer RX buffer.

HEADER

The packet header is comprised of H0, LENGTH, and H1, in that order. Each of the 3 fields of the header has programmable length, from 0 to 16 bits. A size of zero for any of the header components infers that that component is skipped in the packet structure. Although the size of the individual H0, LENGTH, and H1 components need not be aligned to a byte boundary, the overall header must be byte-aligned. That is, the sum of the sizes (in bits) of H0, LENGTH, and H1, must be a integer multiple of 8 bits. This is a software restriction, a violation of which may result in indeterminate behavior.

For GENERIC_FSK protocol, the purpose of the header is twofold:

1. To frame the LENGTH field, so that LENGTH can be extracted and interpreted by the GENERIC_FSK packet processor, so as to determine how and when perform the required bit-stream processing on the data. In most wireless protocols with regular structures, the LENGTH field is embedded within the packet header, but its positioning, number of LENGTH bits, bit ordering, and interpretation can differ. The GENERIC_FSK configurability thus provides support for a wide range of header formats, yielding a high likelihood that most current and future packet formats and LENGTH fields will be parsable by the GENERIC_FSK Link Layer controller.
2. To provide rudimentary packet filtering on the non-LENGTH header bits, namely H0 and H1. Screening of packets based on H0 and/or H1 bit comparisons can reduce power consumption, by avoiding notification to the host processor on received packets that do not comply with pre-configured H0 and/or H1 bitmasks, enabling longer MCU sleep durations.

The header components, and their configurability options, are further described below.

H0

H0 is an optional header packet element. Its length in bits is determined by register H0_SZ[4:0]. Legal values for H0_SZ are:

0 <= H0_SZ <= 16

If H0 is present (non-zero size), it can be filtered-on by the GENERIC_FSK packet processor; each bit of H0 can be forced to match a programmable bitmask, or ignored. Every bit of H0 that is forced to match the bitmask, must match; otherwise, the packet is rejected. The registers H0_MATCH[15:0] and H0_MASK[15:0] control H0 filtering. For every bit of H0_MASK that is set to one, the received bit in the H0 portion of the header

must match the corresponding bit of H0_MATCH. Bits of H0_MASK which are cleared, do not require a match of received data bits to H0_MATCH bits. If any bit of received data which is qualified with H0_MASK[x]=1 fails to match its corresponding H0_MATCH[x] bit, the H0_FAIL register status bit will be set, an RX recycle will occur, and the host processor will not be notified. (The recycle operation will self-clear the H0_FAIL bit). For debug purposes, the control bit REC_BAD_PKT has been provided. If REC_BAD_PKT=1 and a H0-mismatch occurs, the RX recycle is prevented, the full packet is received, the MCU is notified, and the H0_FAIL flag stays asserted. In this case, the H0-fail flag will stay asserted until the next RX warmup (specifically, rx_init clears the bit).

The equation for H0_FAIL is thus:

$H0\_FAIL = [H0_{received} \text{^} H0\_MATCH[15:0]) \ \& \ H0\_MASK > 0$

LENGTH

LENGTH is an optional header packet element. Its length in bits is determined by the register LENGTH_SZ[4:0]. Legal values for LENGTH_SZ are:

0 <= LENGTH_SZ <= 16

If LENGTH is present (non-zero size), its value determines the number of octets remaining in the packet after the header is complete, that is, PAYLOAD octets plus CRC octets, as depicted below.

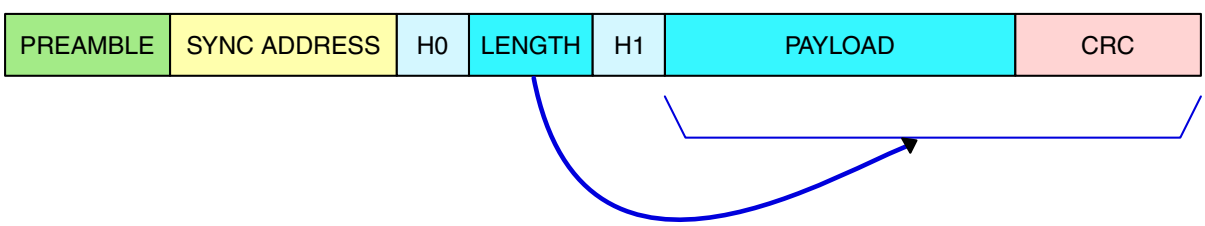


Figure 45-104. LENGTH Field Interpretation with No LENGTH_ADJ

For example, if the payload consists of 16 octets, and CRC is 4 octets, the LENGTH field of the header should be set to 20. The packet processor will extract the packet length from the LENGTH field of the header, and use this information to determine the timing of the CRC, Whitening, and Manchester control signals, as well as end-of-packet interrupt triggering.

In case a protocol exists for which the LENGTH field is to be interpreted differently, a configurability option has been provided to compensate for this. Hypothetically, this could happen, for example, in a proprietary packet structure, in which LENGTH is to be

defined to represent PAYLOAD only, excluding CRC. The register LENGTH_ADJ, allows a fixed offset to be applied to the extracted LENGTH field, such that the sum of LENGTH + LENGTH_ADJ represents the number of PAYLOAD + CRC octets, as depicted below:

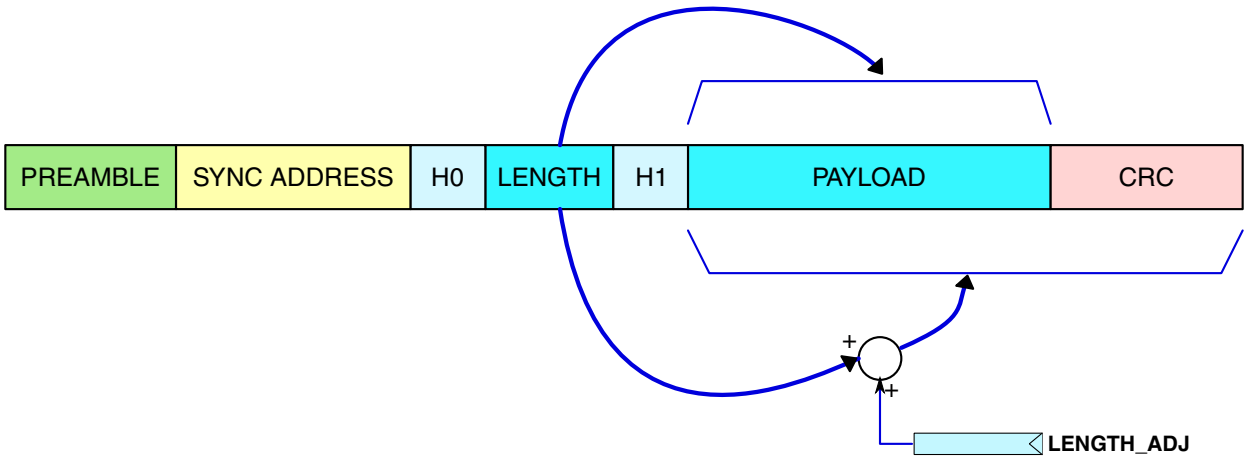


Figure 45-105. LENGTH Field Interpretation with LENGTH_ADJ

Applying the previous example, if, using a proprietary format, LENGTH were to be defined as payload-length only, LENGTH would be set to 16. In order for the packet processor to correctly time the CRC and Whitening control signals, etc., the LENGTH_ADJ register should be programmed to 4 in this case, so that the packet processor would operate on a 20-octet “PAYLOAD + CRC” PDU. The LENGTH_ADJ register represents a signed value, so that negative offsets can be applied. The legal range for LENGTH_ADJ is:

-31 <= LENGTH_ADJ <= 31

The LENGTH field could be ordered MSB first or LSB first. The packet processor must be able to accommodate either case, in order to extract the LENGTH field properly from the header, so a configurability option has been provided. The register bit LENGTH_BIT_ORD specifies the bit order. If LENGTH_BIT_ORD=0, ordering is LSB-first, as shown below.

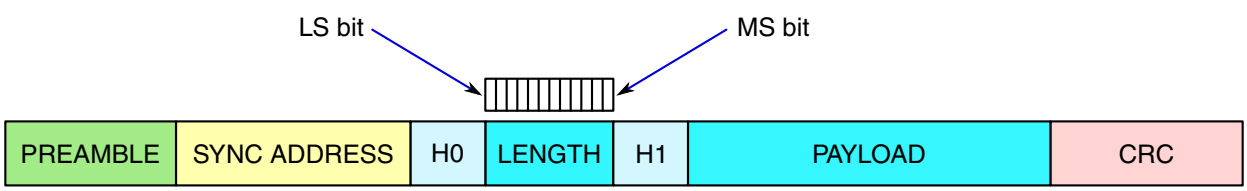


Figure 45-106. LENGTH Bit Ordering (LSB First)

If LENGTH_BIT_ORD=1, ordering is MSB-first, as shown below.

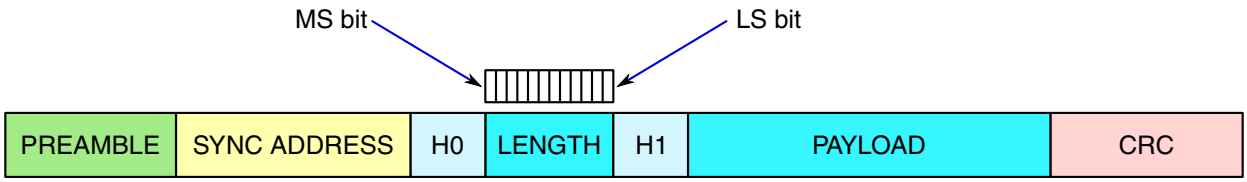


Figure 45-107. LENGTH Bit Ordering (MSB First)

If LENGTH_SZ=0, then LENGTH is not present in the header. The packet processor interprets this as LENGTH=0. If, in this case, LENGTH_ADJ=0 also, the packet processor interprets this to mean that the packet ends with the last bit of H1, and there are no PAYLOAD or CRC octets (CRC_SZ should be programmed to 0 in this case). There is no CRC verification in this scenario. If, in this case, LENGTH_SZ=0, meaning an empty payload field, but a CRC check is desired, CRC_SZ should be programmed to the desired number of CRC octets, and LENGTH_ADJ should be programmed to the same value: LENGTH_ADJ = CRC_SZ.

For GENERIC_FSK, valid payload lengths are from 0 to 2047 octets, i.e.,

0 <= LENGTH <= 2047

If LENGTH_ADJ is non-zero, then the maximum limit of 2047 applies to the sum of LENGTH and LENGTH_ADJ, i.e.,

0 <= (LENGTH + LENGTH_ADJ) <= 2047

A restriction on LENGTH_ADJ follows:

If the packet ends immediately after the LENGTH field of the header, i.e., H1_SZ=0 and there is neither payload nor CRC, then LENGTH_ADJ must be set to 0.

Although the LENGTH field has a maximum size of 16 bits, since the maximum receivable packet size is 2047 bytes, any received LENGTH bits above bit [10] are ignored by the packet processor, which effectively clamps LENGTH at 2047 bytes. There may be cases where it is desirable to limit the maximum received packet length to values less than 2047, so that received packets whose extracted LENGTH field indicates to be larger than a pre-set limit, are rejected by the packet processor. A configurability option has been provided for this, the register LENGTH_MAX[6:0]. The LENGTH_MAX setting configures a pre-set limit on the extracted LENGTH field, in multiples of 16 octets. A special case exists for LENGTH_MAX=0 (the register default), which implies no hardware-limiting; this setting should be used to allow reception of full-length 2047-octet packets. If LENGTH_MAX=1, packets with LENGTH >= 16 octets are rejected. If LENGTH_MAX=2, packets with LENGTH >= 32 octets are rejected. If LENGTH_MAX=127 (the maximum setting) packets with LENGTH >= 2032 octets are rejected. To determine compliance with LENGTH_MAX, the packet processor compares

the LENGTH_MAX register with the extracted LENGTH field directly, not the adjusted length (LENGTH + LENGTH_ADJ). If a packet is received during which the extracted LENGTH field exceeds the LENGTH_MAX setting, the LENGTH_FAIL register status bit will be set, an RX recycle will occur, and the host processor will not be notified. (The recycle will self-clear the LENGTH_FAIL bit). For debug purposes, the control bit REC_BAD_PKT has been provided. If REC_BAD_PKT=1 and a LENGTH_MAX violation occurs, the RX recycle is prevented, the full packet is received, the MCU is notified, and the LENGTH_FAIL flag stays asserted. In this case, the LENGTH_FAIL flag will stay asserted until the next RX warmup (specifically, rx_init clears the bit).

H1

H1 is an optional header packet element. Its length in bits is determined by register H1_SZ[4:0]. Legal values for H1_SZ are $0 \leq H1\_SZ \leq 16$. If H1 is present (non-zero size), it can be filtered-on by the GENERIC_FSK packet processor; each bit of H1 can be forced to match a programmable bitmask, or ignored. Every bit of H1 that is forced to match the bitmask, must match; otherwise, the packet is rejected. The registers H1_MATCH[15:0] and H1_MASK[15:0] control H1 filtering. For every bit of H1_MASK that is set to one, the received bit in the H1 portion of the header must match the corresponding bit of H1_MATCH. Bits of H1_MASK which are cleared, do not require a match of received data bits to H1_MATCH bits. If any bit of received data which is qualified with H1_MASK[x]=1 fails to match its corresponding H1_MATCH[x] bit, the H1_FAIL register status bit will be set, an RX recycle will occur, and the host processor will not be notified. (The recycle will self-clear the H1_FAIL bit). For debug purposes, the control bit REC_BAD_PKT has been provided. If REC_BAD_PKT=1 and a H1-mismatch occurs, the RX recycle is prevented, the full packet is received, the MCU is notified, and the H1_FAIL flag stays asserted. In this case, the H1-fail flag will stay asserted until the next RX warmup (specifically, rx_init clears the bit).

The equation for H1_FAIL is thus:

$H1\_FAIL = [H1_{received} \text{\^{}} H1\_MATCH[15:0]) \ \& \ H1\_MASK > 0$

PAYLOAD

In the GENERIC_FSK packet structure, the payload directly followed the header. For transmission, payload bytes are serialized, and transmitted, LSB first, in order, as they are received from the Packet Buffer interface. During reception, payload bytes are received LSB first, undergo serial-to-parallel conversion to octets, 4 octets are packed into a 32-bit word, and the word is transferred to memory via Packet Buffer. For both TX and RX, since the packet processor interprets the extracted LENGTH to represent the number of Payload + CRC octets, the actual number of payload octets is computed from the following equation:

$Number\_of\_Payload\_Octets = (LENGTH_{extracted} + LENGTH\_ADJ - CRC\_SZ)$

For TX, payload bytes are subjected to whitening or Manchester encoding. They are also shifted through CRC to compute a CRC checksum value. Whitening, encoding, and CRC computing are all optional. Whitening and Manchester encoding are mutually exclusive. For RX, payload bytes are subjected to de-whitening, Manchester decoding, and CRC verification. De-whitening, Manchester decoding, and CRC verification are all optional. De-whitening and Manchester decoding are mutually exclusive. For both TX and RX, (de-)whitening can precede CRC, or follow it. Manchester encoding always operates on post-CRC bitstream for transmission, and Manchester decoding always precedes CRC verification for reception. For more details on CRC, whitening, and Manchester encoding, see Section Bitstream Processing.

CRC

In the GENERIC_FSK packet structure, the CRC field directly followed the payload. The size of the CRC field, in octets, is set by the CRC_SZ register. The valid range for CRC_SZ is:

0 <= CRC_SZ <= 4

For both TX and RX, the portion of the packet which are subjected to CRC is programmable, via the register CRC_START_BYTE[3:0]. The register CRC_START_BYTE programs the absolute byte count of the first byte for which CRC shifting is desired. The CRC_START_BYTE value is referenced to the GENERIC_FSK Byte Counter. The Byte Counter starts at 0 for the first octet of Sync Address, and increments octet-for-octet henceforth, until the last octet of CRC is transmitted or received. An example of CRC_START_BYTE programming is shown below.

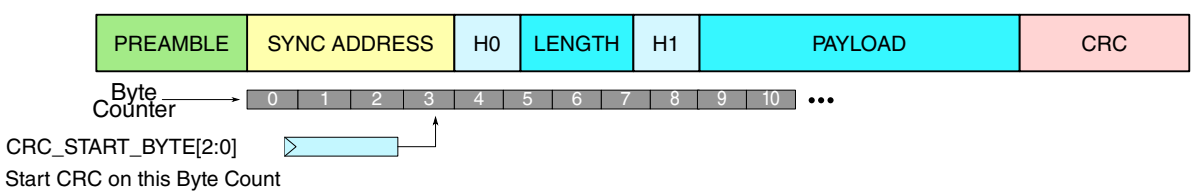


Figure 45-108. CRC Start Byte

In this example

SYNC_ADDR_SZ = 4

Header Size = 4 i.e. (H0_SZ + LENGTH_SZ +H1_SZ)/8 = 4

To initiate CRC shifting at the start of Sync Address, program CRC_START_BYTE = 0

To initiate CRC shifting at the start of Header, program CRC_START_BYTE = 4

To initiate CRC shifting at the start of Payload, program CRC_START_BYTE = 9

The CRC_START_BYTE should not be programmed so large as to exceed the length of the PDU; that is, larger than extracted LENGTH – CRC_SZ. If CRC_SZ=0, program CRC_START_BYTE=0.

For more details on how the GENERIC_FSK packet processor controls the CRC generation and verification processes, see Section Bitstream Processing.

EXAMPLE

A real-world example follows, describing how the GENERIC_FSK Link Layer packet configuration can be applied to transmit and receive packets which conform to IEEE 802.15.4g . For this example, please refer to IEEE Std 802.15.4g™-2012, Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks.

Section 18.1.1.3 of 802.15.4g describes the packet header (PHR), which is replicated below as it appears in that standard:

Table 45-38. Format of the PHR(without mode switching) for MR-FSK

Bit string index	0	1-2	3	4	5-15
Bit mapping	MS	R_1-R_0	FCS	DW	L_{10}-L_0
Field Name	Mode Switch	Reserved	FCS Type	Data Whitening	Frame Length

The GENERIC_FSK Link Layer controller supports such a header format, and can be programmed to transmit and receive packets which conform to it. To correctly size and order the GENERIC_FSK header elements so as to conform with this header format:

1. The LENGTH field ends the header, so there is no H1 (H1_SZ=H1_MASK=0)
2. The 5 bits which precede LENGTH in the header, are grouped into H0 (H0_SZ=5)
3. The size of the LENGTH field is 11 bits (LENGTH_SZ=11)
4. The LENGTH is specified as MSB-first (LENGTH_BIT_ORD=1)

For this example, we would like to leverage the H0-filtering capability built into the GENERIC_FSK Link Layer to screen the 802.15.4g header, to require that the bit fields in the received packet comply with the following conditions:

1. MS (Mode Switch, bit [0] of header) = 0
2. DW (Data Whitening, bit [4]) = 1
3. FCS (FCS Type, bit [3]) = 1 (Four-octet FCS)
4. Bits [1] and [2] are marked as Reserved, to be ignored on reception

Given these parameters, the following diagram shows how we would program the GENERIC_FSK Link Layer controller to accept this packet configuration, and then screen H0 accordingly:

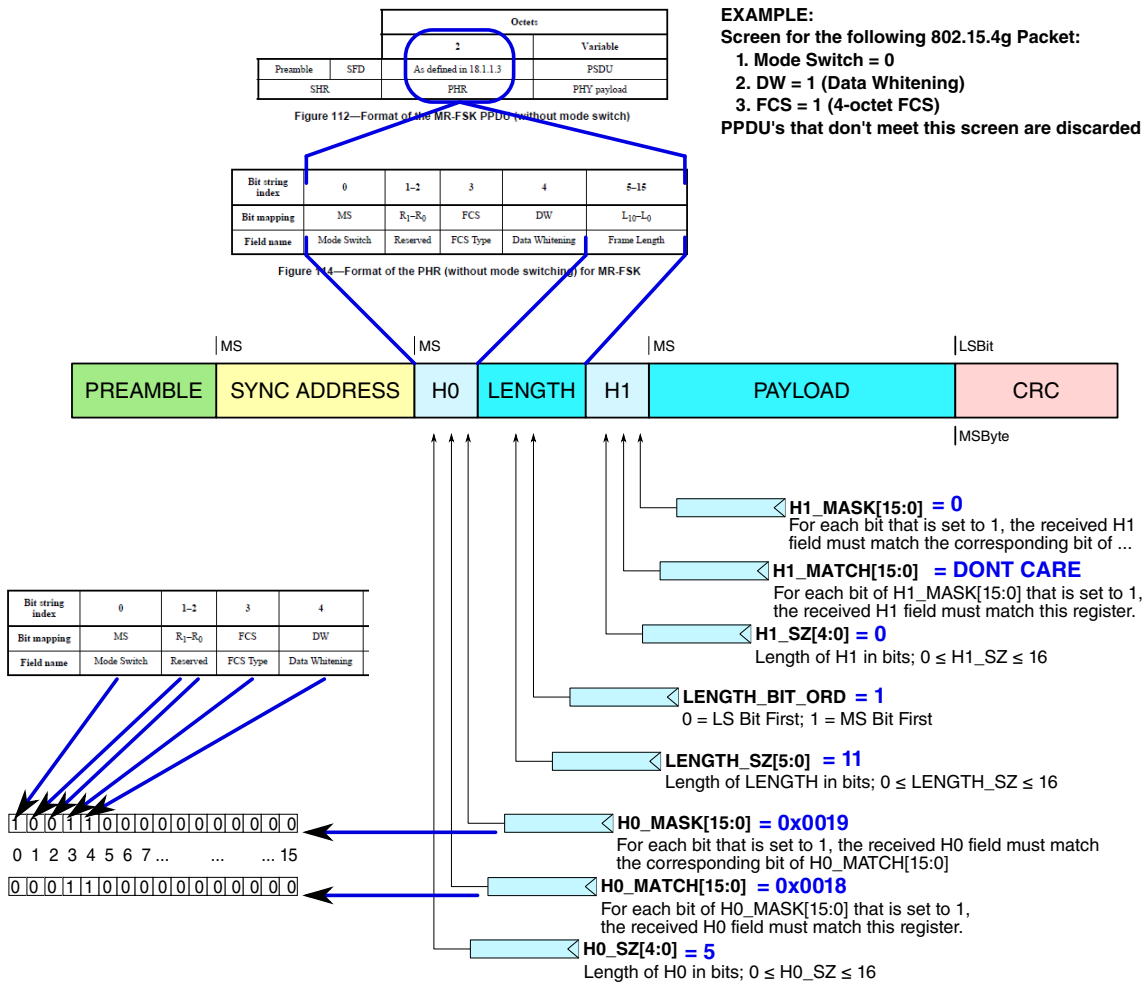


Figure 45-109. GENERIC FSK PACKET STRUCTURE AS APPLIED TO 802.15.4g PHR

Received packets which meet the H0 filtering screen are fully received, and Data Indication is provided to the host processor. Packets which fail the screen result in an RX recycle, and no notification to the host.

45.5.2.3.2 Bitstream Processing

The GENERIC_FSK Link Layer controller provides a number of options to allow for control and modification of the serial bitstream as the packet is being transmitted and received. From a high-level point-of-view, these 4 options are provided:

1. CRC generation (TX) and CRC verification (RX)
2. Data Whitening (TX) and De-whitening (RX)
3. Manchester Encoding (TX) and Decoding (RX)
4. Variable Bit Rate

The following diagram provides another perspective of bitstream processing.

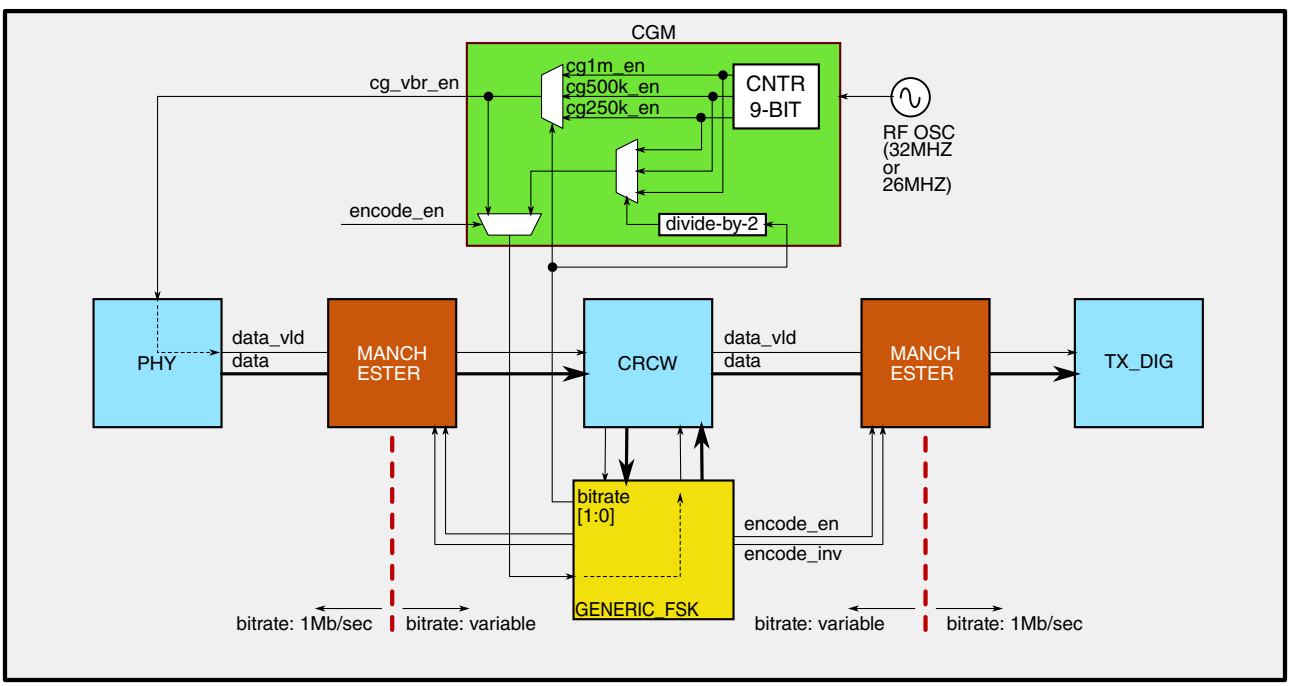


Figure 45-110. BITSTREAM PROCESSING WITH MANCHESTER ENCODING

For GENERIC_FSK, CRC and Whitening are somewhat consolidated, and are performed in a separate module external to the Link Layer Controller. The CRC/Whitener block performs these functions. Manchester encoding is also performed externally, and is described later in this section. Bitrate configurability is managed jointly by the GENERIC_FSK Link Layer controller and the 2.4GHz Radio's Clock Generation Module (CGM). The following sections provide more detail on how CRC, whitening, Manchester encoding, and bitrate, are controlled by the GENERIC_FSK Link Layer, and describes the configuration options associated with each.

CRC & WHITENING

The GENERIC_FSK Link Layer controller dynamically manages the CRC generation process (for TX), and CRC verification process (for RX), by way of 3 timing signals that it asserts to the CRC/Whitener module.

For GENERIC_FSK, the CRC engine in the CRC/Whitener block shall only shift incoming bits (data_in) through its LFSR when all the following conditions are met:

1. crcw_init=0
2. crc_en=1
3. tx_data_in_vld=1 (signal is data_in_vld at the CRC/Whitener)

The CRC/Whitener shall only shift bits out of its CRC LFSR, to the TX digital block, when all the following conditions are met:

1. crcw_init=0
2. crc_comp=1
3. tx_data_in_vld=1 (signal is data_in_vld at the CRC/Whitener)

The packet processor uses the extracted LENGTH field of the packet, as well as the registers which control the size of the individual packet elements, in order to determine when to assert the CRC-related control signals, **crc_en** and **crc_comp**. This is true for both transmission and reception. The calculations used by the packet processor to determine when to assert the CRC-related control signals are summarized in the following diagram.

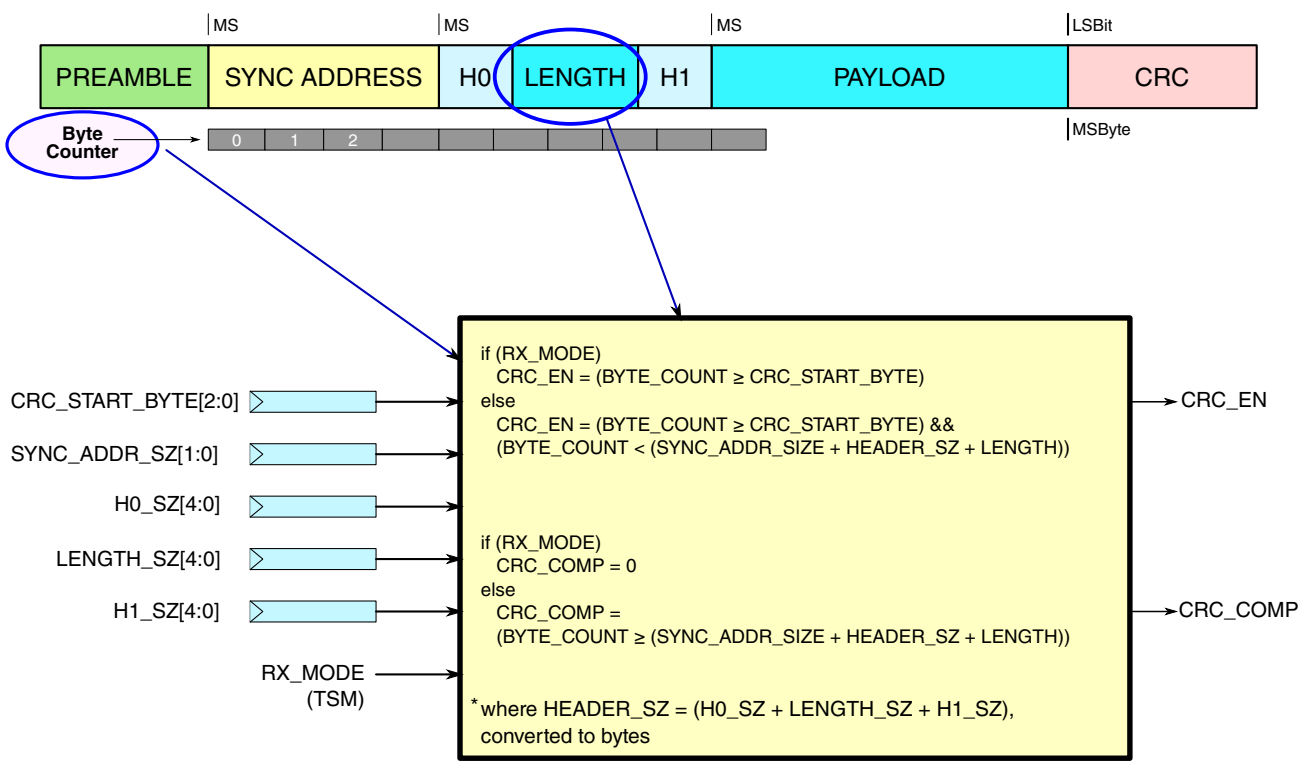


Figure 45-111. GENERIC FSK: CONTROL OF CRC

Likewise, the GENERIC_FSK Link Layer controller optionally enables whitening and de-whitening of the packet bitstream, and controls the portions of the packet which are subjected to whitening. For both TX and RX, the portion of the packet which are

subjected to whitening/de-whitening is programmable; the following table describes the 3 registers which determine which elements of the packet bitstream are whitened/de-whitened.

REGISTER NAME	FUNCTIONAL DESCRIPTION
WHITEN_START[1:0]	**0**: No whitening/de-whitening **1**: Begin whitening/de-whitening at the start of the header (H0) field, regardless of packet length. **2**: Begin whitening/de-whitening at the start of the header H1 field, but only if the extracted length field meets or exceeds WHITEN_SZ_THR, i.e., ($LENGTH_{extracted}$ >= WHITEN_SZ); otherwise, there is no whitening/de-whitening of the bitstream **3**: Begin whitening/de-whitening at the start of the PAYLOAD, but only if the extracted length field meets or exceeds WHITEN_SZ_THR, i.e., ($LENGTH_{extracted}$ >= WHITEN_SZ); otherwise, there is no whitening/de-whitening of the bitstream
WHITEN_END	**0**: Whitening/de-whitening ends with the last bit of the last PAYLOAD byte **1**: Whitening/de-whitening ends with the last bit of the last CRC byte
WHITEN_SZ_THR[12:0]	If WHITEN_START=2 or WHITEN_START=3, apply whitening/de-whitening at the point determined by WHITEN_START, but only if the extracted LENGTH field of the packet meets or exceeds this WHITEN_SZ_THR threshold; if LENGTHextracted < WHITEN_SZ_THR, there is no whitening/de-whitening of the bitstream. If WHITEN_START < 2, this register has no effect.

The GENERIC_FSK Link Layer controller dynamically manages the data whitening process (for TX), and de-whitening (for RX), via 2 timing signals that it asserts to the CRC/Whitener module.

For GENERIC_FSK, the Whitening/de-whitening engine in the CRC/Whitener block shall only shift incoming bits (data_in) through its LFSR when all the following conditions are met:

1. crcw_init=0
2. whiten_en=1
3. tx_data_in_vld=1 (signal is data_in_vld at the CRC/Whitener)

The packet processor uses the extracted LENGTH field of the packet, as well as the registers which control the size of the individual packet elements, in order to determine when to assert the whitening-related control signal, whiten_en. This is true for both transmission and reception. The calculations used by the packet processor to determine when to assert whiten_en control signals are summarized in the following diagram.

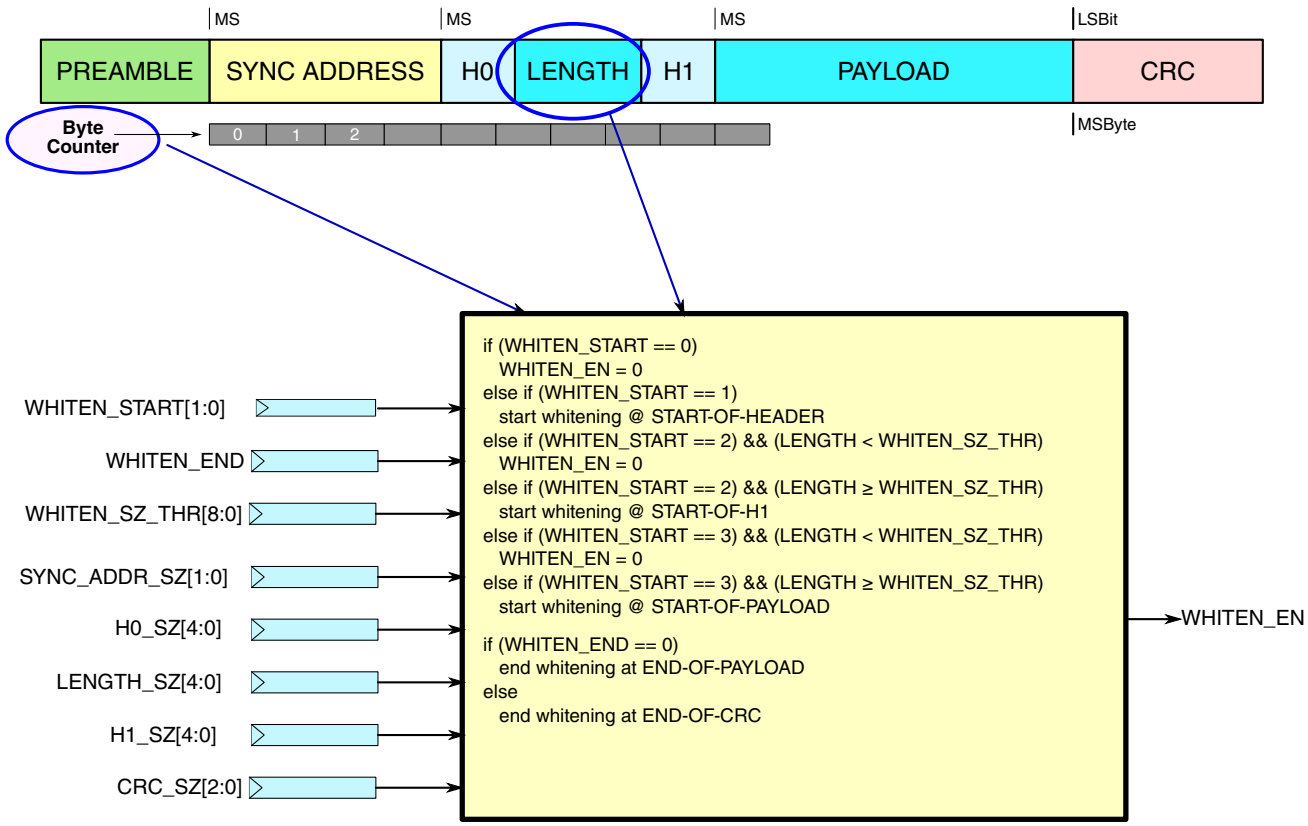


Figure 45-112. GENERIC FSK: CONTROL OF WHITENING

The order in which CRC and whitening are performed, is configurable, via the static register bit WHITEN_B4_CRC. This bit is programmed in the GENERIC_FSK Link Layer, and sent to the CRC/Whitening engine to control the ordering. The following table describes how the bitstream is processed for both settings of WHITEN_B4_CRC, for both TX and RX.

	TX	RX
WHITEN_B4_CRC=1	In the CRC/Whitener, the CRC engine shifts in data bits which have been whitened first through the whitener LSFR. The entire post-whitened bitstream is then transmitted to the TX Digital.	Since the entire bitstream, including CRC, has been pre-whitened, the incoming bitstream from the PHY is shifted directly through the CRC LFSR to verify the checksum. The entire bitstream is simultaneously de-whitened by the whitener LFSR and shifted out to the Link Layer.

Table continues on the next page...

	TX	RX
WHITEN_B4_CRC=0	In the CRC/Whitener, the CRC engine shifts in data bits which have not been whitened. Whitening is performed on the CRC LFSR output. The entire post-whitened bitstream is then transmitted to the TX Digital.	The entire bitstream is de-whitened first by the whitener LFSR, and then shifted through the CRC LFSR to verify the checksum. The entire de-whitened bitstream is simultaneously shifted out to the Link Layer.

A diagram depicting an example of bitstream processing flow during packet transmission, with WHITEN_B4_CRC=1 (the register default), is shown below.

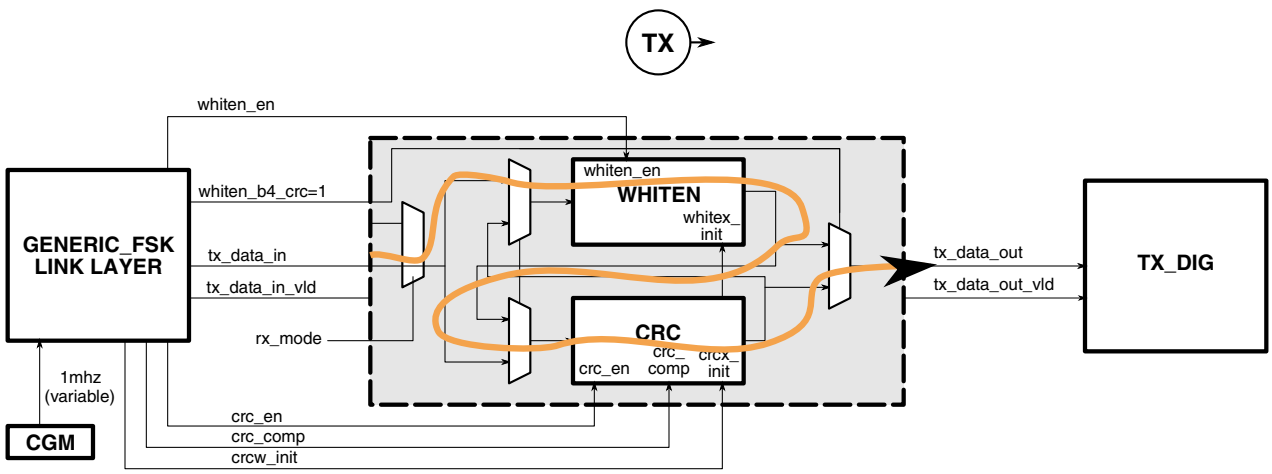


Figure 45-113. Whitening Before CRC (TX)

A diagram depicting an example of bitstream processing flow during packet reception, with WHITEN_B4_CRC=1, is shown below.

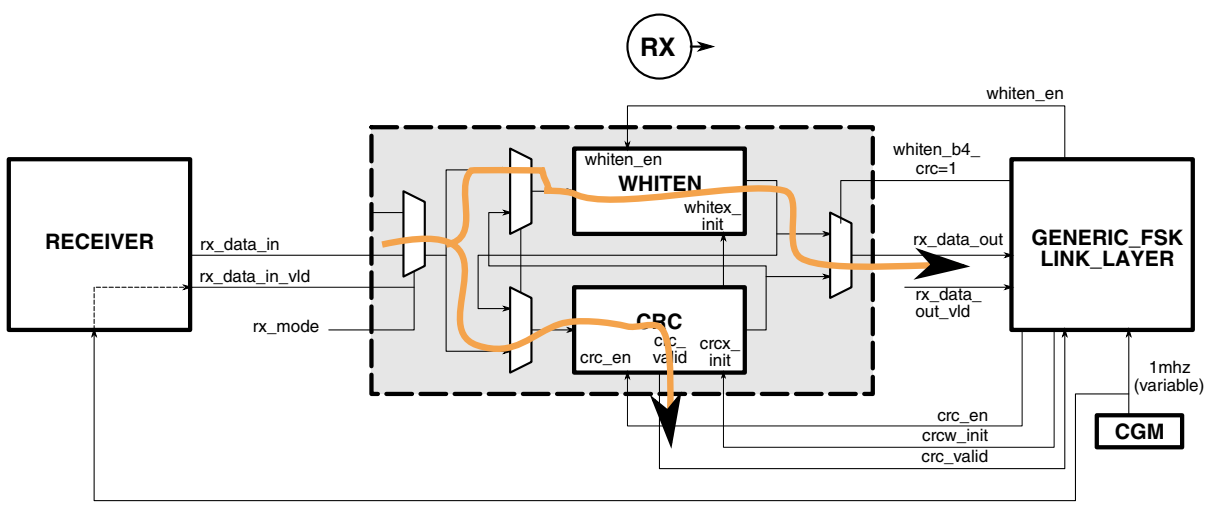


Figure 45-114. Whitening Before CRC (RX)

A timing diagram depicting a real-world example of TX bitstream processing timing is shown below. In this example, the desired start point for CRC shifting is the start of Network Address, so CRC_START_BYTE=0. The desired start point for whitening is the start of the header (H0), so WHITEN_START=1. It is also desired to stop whitening after the last bit of the last PAYLOAD byte, so WHITEN_END=0.

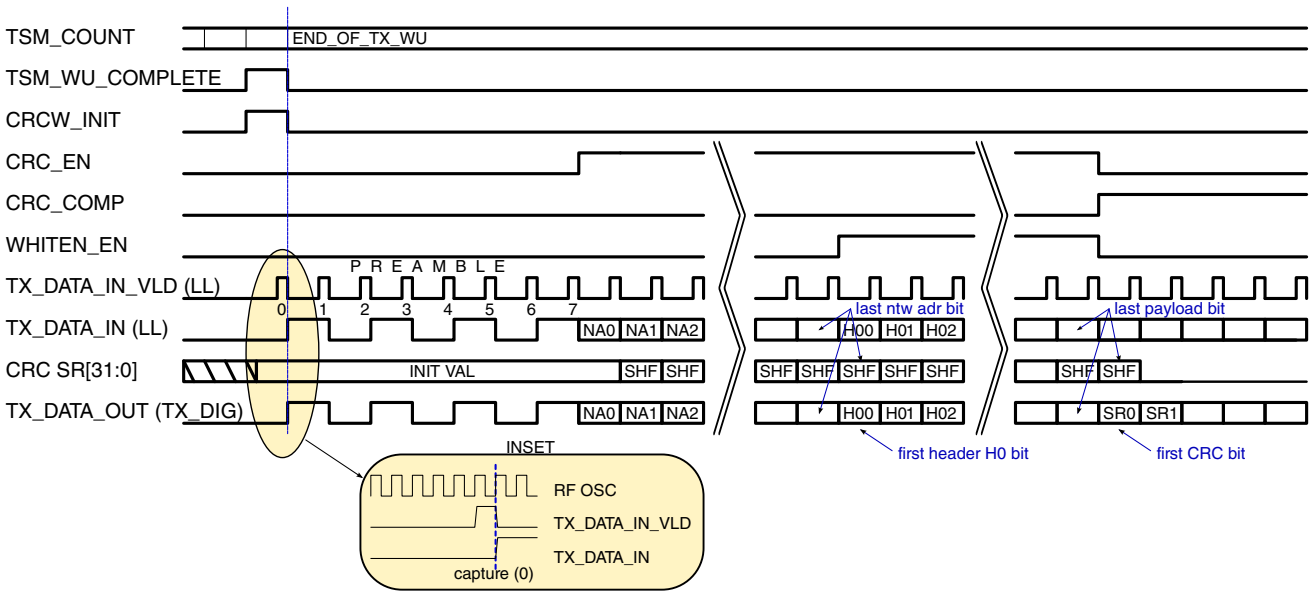


Figure 45-115. BIT STREAM PROCESSING CRC AND WHITENING TX TIMING

The bitstream depicted in the previous timing diagram, with CRC and whitening applied, is shifted out to the TX digital block, and transmitted over the air. A receiving device, will receive and process this same bitstream, by applying de-whitening, and verifying CRC on the de-whitened bitstream. The CRC and whitening parameters (registers) would be programmed to the receiving device, identically to how they are programmed for the transmitting device, specifically:

CRC_START_BYTE=0

WHITEN_START=1

WHITEN_END=0

A similar timing diagram depicting this RX bitstream processing timing, is shown below.

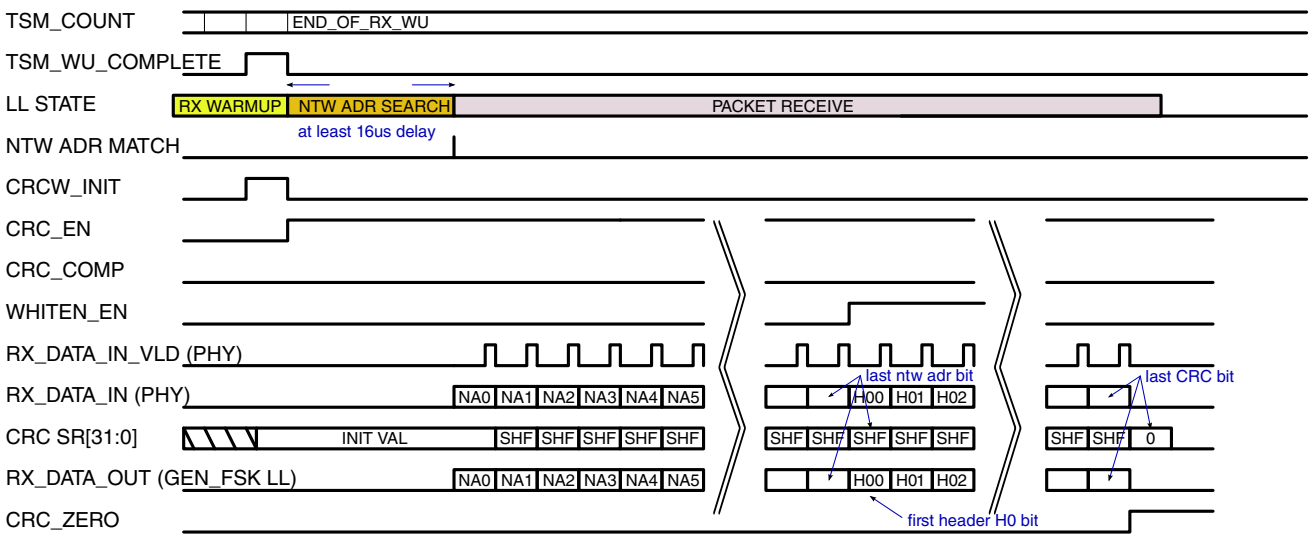


Figure 45-116. BIT STREAM PROCESSING CRC AND WHITENING RX TIMING

The GENERIC_FSK Link Layer controller provides a special whitening option, that causes the whitening LFSR to be re-initialized at the start of the PAYLOAD. The register bit WHITEN_PAYLOAD_REINIT, when 1, will result in the LFSR being re-initialized to its programmed initial state, determined by WHITEN_SEED[8:0], coincident with the first bit of the first byte of PAYLOAD. This applies to both TX and RX. When WHITEN_PAYLOAD_REINIT=0, there is no such re-initialization. This feature is only relevant when whitening/de-whitening is programmed to start before the PAYLOAD, i.e., WHITEN_START=1 or WHITEN_START=2.

The GENERIC_FSK Link Layer controller provides a number of other configurability options, which affect the internal operation of the CRC engine, and not the Link Layer controller itself. A listing of these registers which affect CRC internal operation, appears in the table below. For more details on how these registers affect CRC operation, see the CRC/Whitener Chapter.

CRC-related Register	Brief Description
CRC_POLY[31:0]	CRC Polynomial 1: XOR exists in the bit's feedback path 0: no XOR in the bit's feedback path
CRC_SEED[31:0]	CRC Initialization Value
CRC_BYTE_ORD	0: LS Byte First 1: MS Byte First
CRC_REF_IN	0: do not manipulate input data stream 1: reflect each byte in the input stream bitwise
CRC_REF_OUT	0: do not manipulate CRC result

Table continues on the next page...

CRC-related Register	Brief Description
	1: CRC result is to be reflected bitwise (operated on entire word)
CRC_XOR_OUT[31:0]	XOR mask for CRC result (for no mask, should be 0)

Similarly, there are a number of configurable whitening options provided by the Link Layer controller, which affect the internal operation of the whitening engine, and not the Link Layer controller itself. A listing of these registers which affect whitener internal operation, appears in the table below. For more details on how these registers affect whitener operation, see the CRC/Whitener Chapter.

Whitener-related Register	Brief Description
WHITEN_ POLY[8:0]	Polynomial Value for Whitening/De-whitening
WHITEN_POLY_TYPE	Whiten polynomial type. A Fibonacci type LFSR is used with the whiten polynomial if this bit is asserted. Otherwise, a Galois type LFSR is used.
WHITEN_SEED[31:0]	Initialization Value for Whitening/De-whitening
WHITEN_SIZE[3:0]	Whitener Length
WHITEN_REF_IN	The input data stream is reflected, bit-wise, per byte, if this bit is asserted. Bit 7 becomes bit 0, bit 6 becomes bit 1, etc. This bit will only cause the reflection of the payload data bits as they are used in the whiten calculation and will not cause any change in the output bit order.

BITRATE

The GENERIC_FSK Link Layer controller provides multiple bitrate options. The register BITRATE[1:0] selects the bitrate, according to the following table.

BITRATE[1:0] Register	BITRATE
0	1 Mbit/sec
1	500 Kbit/sec
2	250 Kbit/sec
3	Reserved

For transmission, BITRATE affects the rate at which the Link Layer controller shifts out bits, through the CRC/Whitener, and ultimately to the TX Digital Block. For reception, BITRATE affects the rate at which the PHY shifts out bits, through the CRC/Whitener, and ultimately to the Link Layer controller, which receives and processes the bitstream. The Link Layer is the originator of the packet bitstream for transmission; the PHY is the originator of the bitstream for reception. The Link Layer controller determines the bitrate by sending the **bitrate[1:0]** signal to the CGM (Clock Generation Module). The CGM decodes this signal and adjusts the rate at which pulses appear on **cg_vbr_en** to track the

programmed BITRATE. (Note: the 'vbr' in **cg_vbr_en** is an acronym for ‘variable bit rate’.) The CGM divides down the reference oscillator frequency to generate the frequency of the desired BITRATE, and then sends out pulses on cg_vbr_en which are one reference oscillator clock wide, at that rate. The **cg_vbr_en** shall be used as a gating signal (clock gating and/or data gating) inside the blocks which originate the bitstream (Link Layer, PHY), and those that consume it (CRC/Whitener, TX digital). The timing on cg_vbr_en manifests itself on the respective *_data_in_vld and *_data_out_vld signals to and from the 4 blocks that operate on the bitstream. This way, all 4 blocks operate in unison at all times at a single, desired bitrate.

The following diagram depicts how bitrate is controlled by the Link Layer and CGM, and how the respective blocks involved in bitstream processing stay in sync using **cg_vbr_en**.

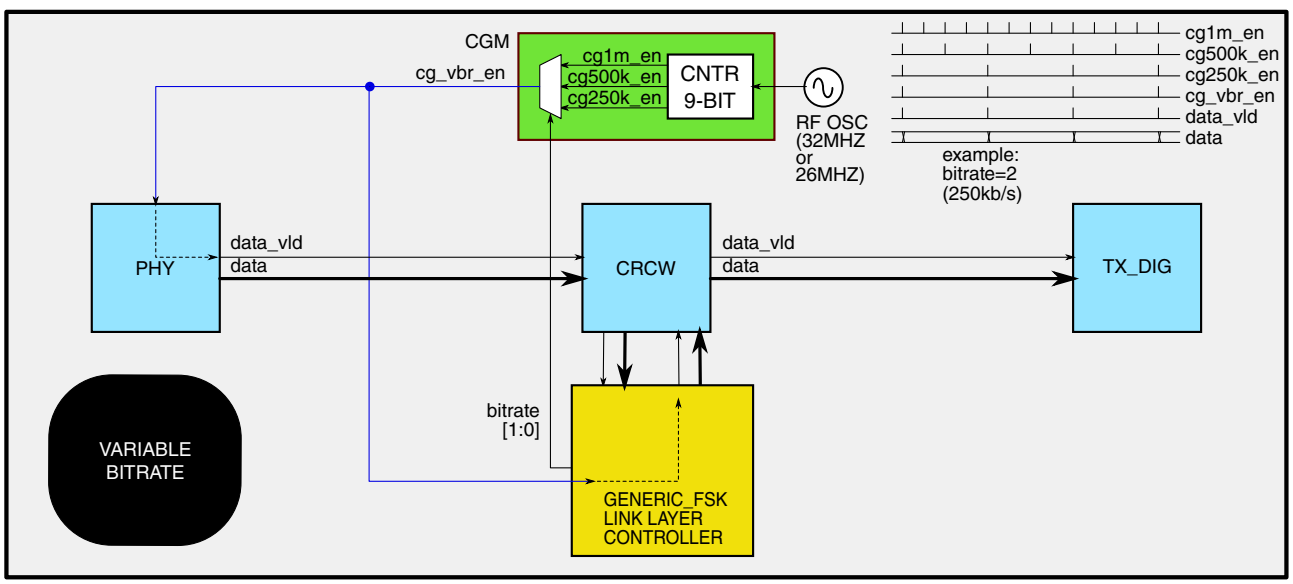


Figure 45-117. BITSTREAM PROCESSING WITH VARIABLE BIT RATE

MANCHESTER ENCODING AND DECODING

The GENERIC_FSK Link Layer controller allows the packet bitstream to be Manchester-encoded for transmission, and decoded for reception. The Manchester encoding used by the controller conforms to IEEE 802.3, which is shown in the following diagram.

Data	Manchester (IEEE 802.3)	Manchester (Inv)
0	10	01
1	01	10

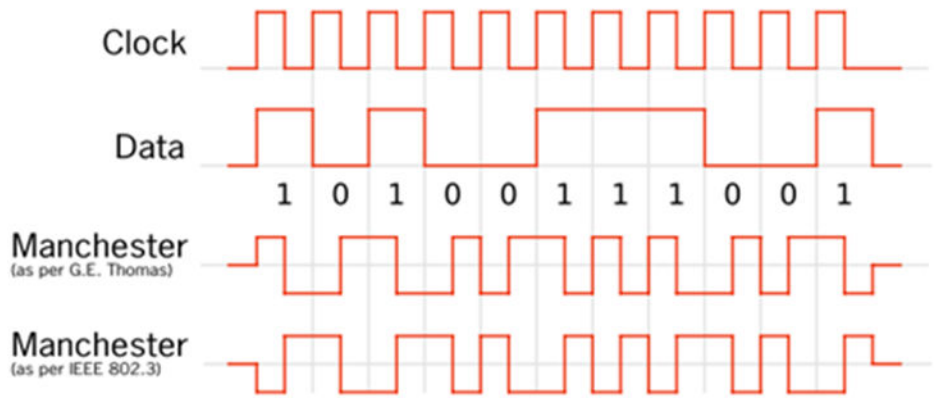


Figure 45-118. Manchester Encoding

Three programmable registers enable and control Manchester encoding and decoding, as described in the following table.

	TX	RX
MANCHESTER_EN	1: the entire packet bitstream, including CRC, is Manchester-encoded as it is transmitted to the TX digital block. CRC operates on the original, non-encoded bitstream. 0: No encoding applied	1: the entire packet bitstream, including CRC, is Manchester-decoded, and the output of the decoder is transmitted simultaneous to the CRC engine and the Link Layer. 0: No decoding applied
MANCHESTER_INV	1: Encoding is per 802.3, but with the coding signal inverted 0: Encoding is as per 802.3	1: Decoding is per 802.3, but with the coding signal inverted 0: Decoding is as per 802.3
MANCHESTER_START	1: Begin encoding at start-of-header 0: Begin encoding at start-of-payload	1: Begin decoding at start-of-header 0: Begin decoding at start-of-payload

As the encoding diagram implies, the post-encoded process bitstream has a bitrate of twice that of the pre-encoded data. The over-the-air bitrate must, of course, be constant. Thus, for transmission, the source of the bitstream data, i.e., the GENERIC_FSK Link Layer controller, must halve its bitrate whenever Manchester encoding is engaged. Since the CRC engine is also upstream of the encoder, it will also see the half bitrate. The output of the Manchester encoder, which feeds the TX digital block, will see the full bitrate. For reception, the source of the bitstream data, i.e, the PHY, will see the full

bitrate. After the PHY feeds the Manchester decoder with the full-rate data, the blocks downstream of the decoder, i.e., the Link Layer controller and the CRC engine, will need to operate at half bitrate. When Manchester encoding is used, the over-the-air bitrate, that is, the bitrate programmed via the BITRATE register, is restricted to either 1Mbit/sec (BITRATE=0), or 500Kbit/sec (BITRATE=1); other BITRATE options are not supported with Manchester. The following table summarizes the supported over-the-air bitrates, and specifies the effective bit rates seen by the 4 blocks involved in bitstream processing while Manchester is engaged.

	PHY	GENERIC_FSK Link Layer	CRC LFSR	TX Digital
BITRATE=0 (ota bitrate = 1 Mbit/sec)	1 Mbit/sec	500 Kbit/sec	500 Kbit/sec	1 Mbit/sec
BITRATE=1 (ota bitrate = 500Kbit/sec)	500 Kbit/sec	250 Kbit/sec	250 Kbit/sec	500 Kbit/sec

Because the Manchester encoding and decoding is not performed over the entire packet, e.g., the Network Address is never encoded, the Link Layer controller must switch its bitrate *dynamically*, reducing its own bitrate by half whenever Manchester is engaged, and restoring full bitrate when Manchester is disengaged. Since the PHY and TX digital must process bits at the full rate at all times, the bitstream-pacing signal, **cg_vbr_en**, must be split, such that the PHY and TX digital get the full rate gating signal, and the Link Layer and CRC get the variable rate gating signal. The following diagram describes how the CGM generates a special half-rate rendition of **cg_vbr_en** when Manchester is engaged (**encode_en**=1), and feeds this to the Link Layer (and indirectly, the CRC engine). The diagram assumes the default setting of BITRATE=0 (over-the-air rate is 1Mb/sec).

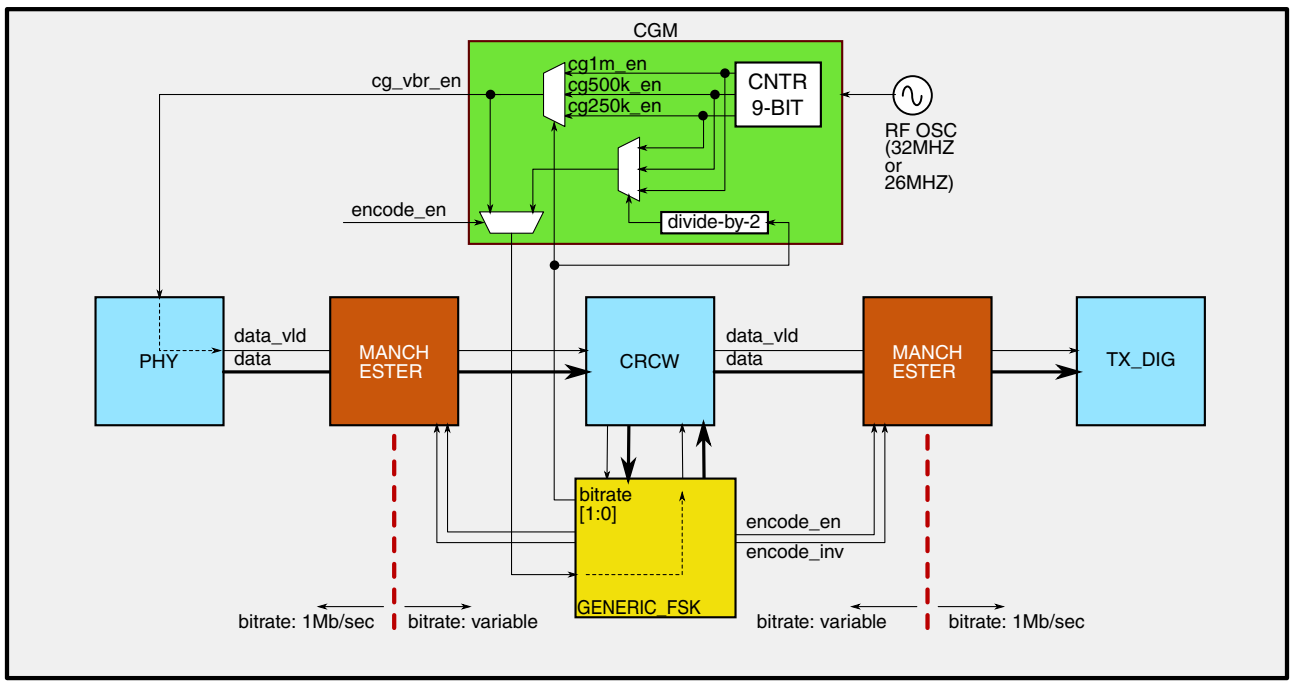


Figure 45-119. BITSTREAM PROCESSING WITH MANCHESTER ENCODING

A timing diagram depicting an example of bitstream processing, including Manchester encoding, for a packet transmission, is included below. The diagram shows the mid-packet switching of the encoding-enable signal, encode_en, and the halving of the bitrate as seen by the Link Layer controller. This example uses MANCHESTER_START=0, to initiate encoding at the start-of-payload.

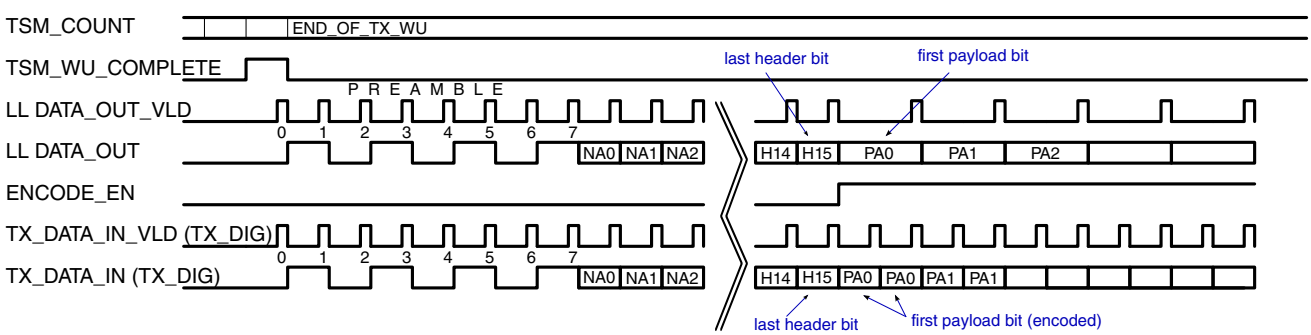


Figure 45-120. BIT STREAM PROCESSING (WITH MANCHESTER) TX TIMING -- 1MBIT/SEC OTA

An example of bitstream processing, including Manchester encoding, for a packet reception, is included below. The diagram shows the mid-packet switching of the decoding-enable signal, encode_en, and the halving of the bitrate as seen by the Link Layer controller. This example uses MANCHESTER_START=0, to initiate encoding at the start-of-payload.

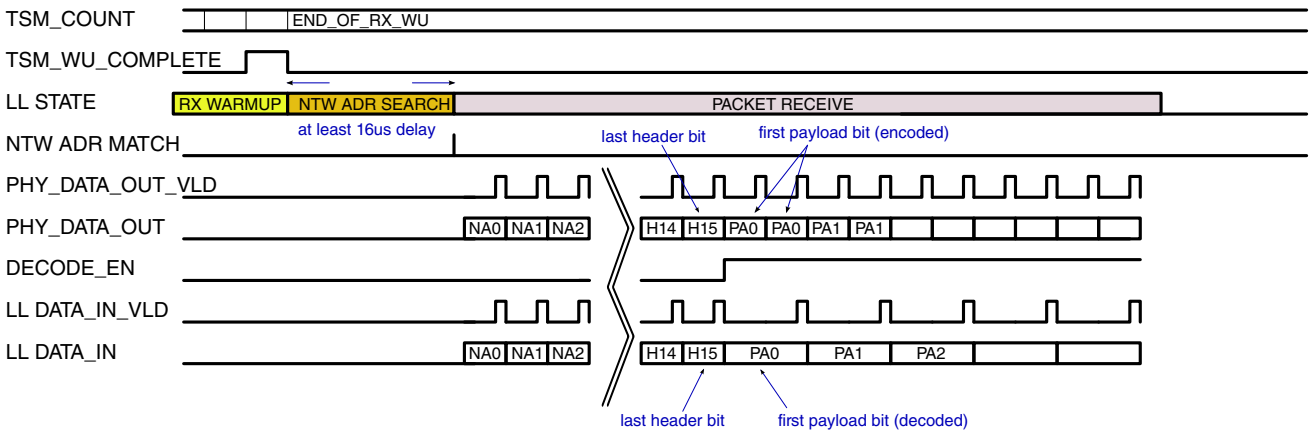


Figure 45-121. BIT STREAM PROCESSING (WITH MANCHESTER) RX TIMING -- 1MBIT/SEC OTA

NOTE

Whitening and Manchester encoding are mutually exclusive, and must not be enabled simultaneously. This is a software restriction on the setting of the register bits WHITEN_START and MANCHESTER_EN. Both may not be simultaneously non-zero. Failure to heed this restriction may lead to indeterminate behavior.

45.5.2.3.3 Fixed Packet Length

The GENERIC_FSK Link Layer controller expects a stream of consecutive bits specifying the length of the packet, to be transmitted and received over-the-air, as part of the packet bitstream, with the length specification typically encoded into the packet header, as shown below.

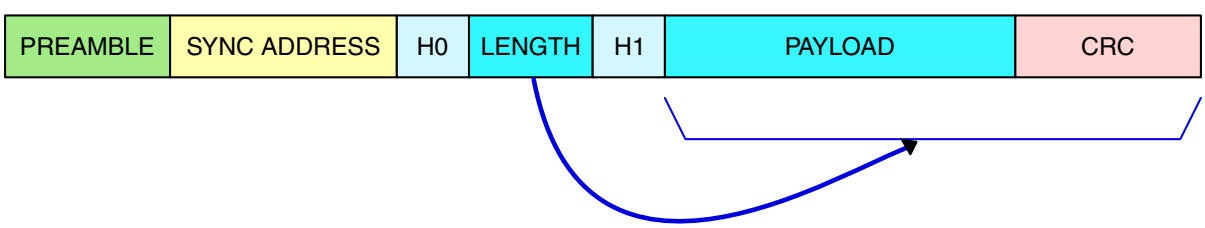


Figure 45-122. PACKET STRUCTURE WITH LENGTH FIELD IN HEADER

Some protocols don't include any length instructions in the packet header, as shown below.

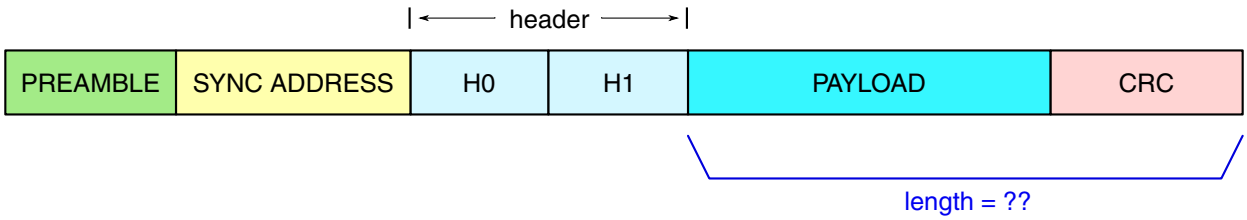


Figure 45-123. PACKET STRUCTURE WITH NO LENGTH FIELD IN HEADER

And, some protocols may not include a header at all.

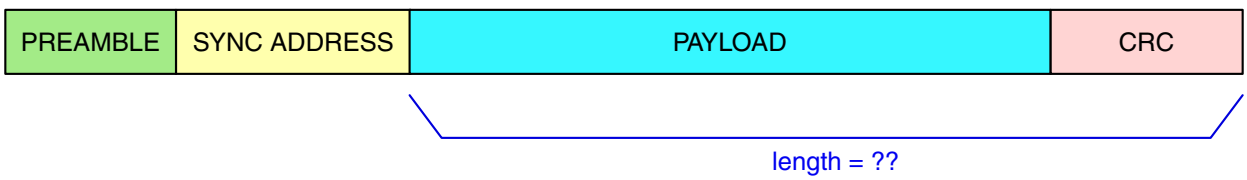


Figure 45-124. PACKET STRUCTURE WITH NO HEADER

In the aforementioned cases, another method of specifying length to the GENERIC_FSK Link Layer hardware must be employed. In both cases, if the length is not specified as part of the over-the-air bitstream, then GENERIC_FSK software must know the length in advance of the TX or RX operation, or know how to calculate it. So GENERIC_FSK software will provide this information to the hardware, in advance of the TX or RX operation, by programming the following registers:

- PACKET_CFG[LENGTH_SZ] = 0 : instructs the hardware that there is no LENGTH info in the packet header, and thus forces the hardware to use LENGTH_ADJ for packet length
- PACKET_CFG[LENGTH_ADJ] = ??? : program the desired (known) length of PAYLOAD + CRC
- PACKET_CFG[LENGTH_ADJ_UNSIGNED] = 1 : allows programmed length up to 255 octets
- CRC_CFG[CRC_SZ] = ??? : program the desired number of CRC octets

As an example, a packet is to be transmitted or received using the GENERIC_FSK Link Layer hardware. The packet has no header, but the length of the payload is known in advance to be 57 octets, and the number of CRC octets is 3. In this case, LENGTH_ADJ is programmed to 60 (57 + 3) and CRC_SZ is programmed to 3, as shown below.

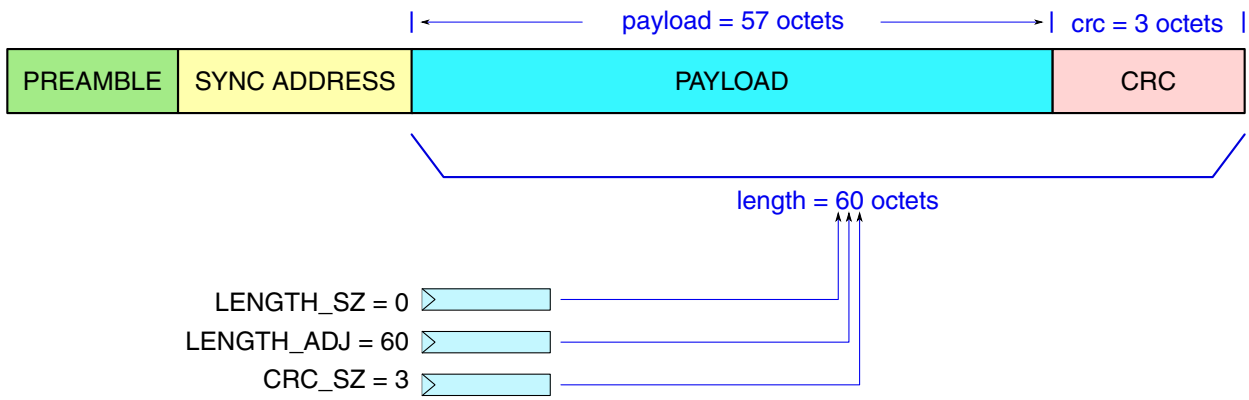


Figure 45-125. PACKET STRUCTURE WITH NO HEADER AND FIXED PACKET LENGTH

45.5.2.3.4 Packet Storage

The GENERIC_FSK Link Layer has access to a Packet Buffer (also known as Packet RAM), to store data to be transmitted, and to receive incoming packet data. The GENERIC_FSK Link Layer software prepares data to be transmitted, by loading the octets, in order, into the Packet Buffer. After reception, the octets received over the air are stored into the Packet Buffer, in the order they are received. The Packet Buffer contains separate spaces for TX and RX data, so incoming receive octets don't overwrite previously-loaded content intended for transmission. The Packet Buffer is large enough to accommodate the longest GENERIC_FSK packet, 2047 octets, plus header and CRC octets, for either transmit or receive.

The GENERIC_FSK Packet Buffer is contained within the multi-protocol Packet RAM. When operating in GENERIC_FSK mode, the entire Packet RAM is allocated to the GENERIC_FSK packet buffer. Because of the potentially large size of the GENERIC_FSK packets, and taking into account the high likelihood of asymmetry between the required lengths of packets for TX and RX, the 2 normally-separated RAM components of the Packet RAM are adjoined sequentially to form a double-depth (1088-entry) single storage space, still 16-bits wide. In addition, the consolidated RAM can be dedicated entirely to TX, entirely to RX, or split between TX and RX. The partitioning of the consolidated RAM between TX and RX is controlled by a configurable TX/RX Partition Point. A Partition Point register, PB_PARTITION[10:0], has been provided. The Register can be programmed with any value between 0 and 1087. The consolidated RAM is partitioned such that the TX buffer resides before the Partition Point (RAM entries 0 -> (Partition_Point-1)), and the RX buffer resides after it (RAM entries Partition_Point -> 1087). Programming the Partition Point register to 0 dedicates the entire consolidated RAM to RX; programming the Partition Point register to 1088 dedicates the entire consolidated RAM to TX; programming the Partition Point register to any value between 1 and 1087 yields a split TX/RX buffer. Separate TX and RX buffers

within the consolidated RAM means that incoming RX packets don't overwrite TX packet data. Because the Packet RAM is 16 bits wide, the GENERIC_FSK Link Layer controller hardware will address the consolidated RAM in an interleaved fashion as it stores octets to, or fetches octets from, the RAM-based packet buffer.

The partitioning of the consolidated RAM between transmit and receive sections for GENERIC_FSK is shown in the diagram below:

1. RAM0 and RAM1 abutted sequentially to form 1 contiguous space from Entry #0 - #1087
2. A Programmable Partition Point is defined: TX data before the partition, RX data after
3. The partition point can be programmed to any value from Entry #0 - #1087
4. Full 2 KB packet can be stored, for TX or RX
5. TX and RX packets are separated (RX does not overwrite TX)
6. Host access restricted to 16-bit

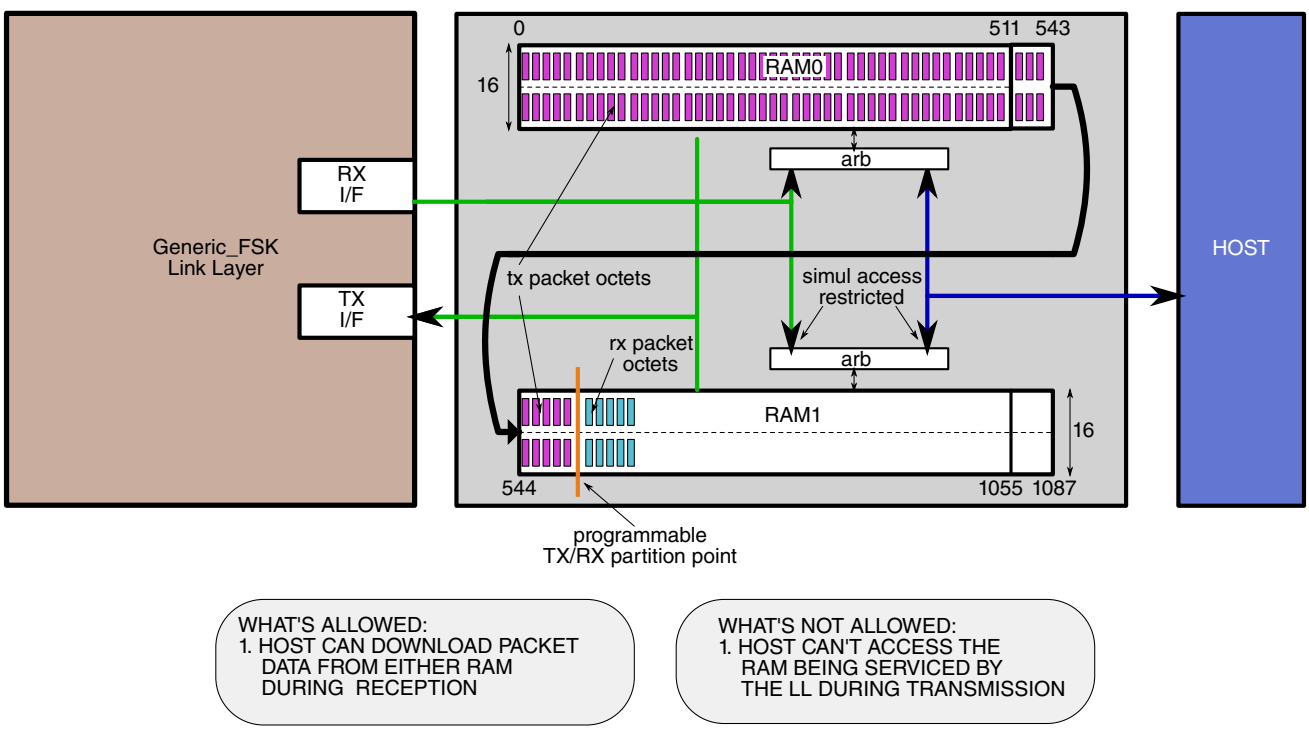


Figure 45-126. RAM ALLOCATIONS FOR GENERIC_FSK

During packet reception, the MCU is allowed to access the RX packet buffer of the consolidated RAM, in order to download the packet while it's being received, if so desired. In cases where both the MCU and the GENERIC_FSK Link Layer controller attempt to simultaneously access the RAM during reception, internal hardware arbitration is provided to delay the MCU access until the GENERIC_FSK Link Layer completes its

access to the RAM. Link Layer accesses complete in a single reference oscillator clock period, so one and only 1 wait state shall be inserted on the IPS bus whenever access attempts coincide.

During packet transmission, MCU access to the consolidated RAM is not allowed. The GENERIC_FSK software must program the TX buffer before commanding a TX operation, and must not access the RAM during the transmission.

As mentioned previously, incoming receive data lands in the RX section of the Packet Buffer. Nominally, a newly received packet will overwrite the contents of the previously received packet (the TX section of the Packet Buffer remains intact). Software can inhibit receive-packet overwriting of the RX section of the Packet Buffer contents by setting the PB_PROTECT bit in the PACKET_RAM_CTRL register, in XCVR address space. When this bit is set, incoming receive data will be blocked from overwriting the existing contents of the RX section of the Packet Buffer. This will block RX packet overwriting, but will not inhibit TX content loading of the TX section of the Packet Buffer.

In the 2.4GHz Radio, the GENERIC_FSK Link Layer is the controlling Link Layer whenever the XCVR_CTRL[PROTOCOL] register is set to 8, 9, or 10. GENERIC_FSK software must not attempt to access the Packet RAM unless XCVR_CTRL[PROTOCOL] is set to 8, 9, or 10. When XCVR_CTRL[PROTOCOL] is set to one of the following: {0, 1, 2, 3, 4, or 5}, another Protocol Engine besides GENERIC_FSK, has exclusive access to the Packet RAM. That also means that the IPS bus assigned to that Protocol Engine, is the only IPS bus allowed to access the Packet RAM. If the GENERIC_FSK Link Layer software attempts to access the Packet RAM via the GENERIC_FSK IPS bus, when XCVR_CTRL[PROTOCOL] is set to one of the above settings, the access attempt will fail, as described below:

1. RAM contents shall not be altered on a GENERIC_FSK write attempt
2. Readback data on a GENERIC_FSK read attempt shall be indeterminate
3. The **ips_xfr_err** shall be asserted on the GENERIC_FSK IPS bus

If XCVR_CTRL[PROTOCOL] is set to neither {8, 9, or 10}, nor {0, 1, 2, 3, 4, or 5}, this means no Protocol Engine has exclusive access to the Packet RAM. GENERIC_FSK software should not attempt to access the Packet RAM under such conditions; results are indeterminate.

45.5.2.3.5 Timebase

The GENERIC_FSK Link Layer includes a dedicated, 24-bit multi-purpose Event Timer (EVENT_TMR), that is updated at a 1 MHz rate. The EVENT_TMR maintains the timebase for all GENERIC_FSK-related activities, except during intervals of Deep Sleep Mode. The clock source for the EVENT_TMR is a high-precision, crystal-referenced RF

Oscillator, which can be either 32MHz or 26MHz. The RF Oscillator is divided down to 1MHz for GENERIC_FSK in the Clocks Generation Module (CGM). To reduce power consumption to a minimum, most of the GENERIC_FSK Link Layer Controller runs off this 1MHz clock. The exceptions are the Register IP bus interface, Command Decoder, and Interrupt Control Unit, parts of which must run at the IP bus frequency.

The 24-bit EVENT_TMR running at 1 MHz rolls over (wraps around) approximately every 16.8 seconds.

The EVENT_TMR count can be read at any time, via the register of the same name. A 32-bit read is recommended to ensure coherency of the 3 bytes that make up the counter. No synchronization to the IP bus is required.

If the EVENT_TMR needs to be updated (changed), this can be done by writing the new, desired EVENT_TMR value to the register of the same name, and setting the EVENT_TMR_LD bit to 1. The update to EVENT_TMR will take effect at the next 1MHz clock edge. The EVENT_TMR[23:0] and EVENT_TMR_LD are all part of the same 32-bit register, and loading of the EVENT_TMR using this method should be performed as an atomic, 32-bit write.

If it is desired to increment the EVENT_TMR by a known amount, this can be done by writing the desired increment value to the EVENT_TMR register, and setting the EVENT_TMR_ADD bit to 1. The increment value is 24-bit signed, thus enabling a decrement function as well. The update to EVENT_TMR will take effect immediately. The EVENT_TMR_ADD avoids the necessity to perform an read-modify-write operation to the EVENT_TMR to adjust its value; this facilitates accurate restoration of the EVENT_TMR state after a period of Deep Sleep Mode. The EVENT_TMR[23:0] and EVENT_TMR_ADD are all part of the same 32-bit register, and incrementing of the EVENT_TMR using this method should be performed as an atomic, 32-bit write.

Two 24-bit timer-compare registers are provided, T1_CMP and T2_CMP.

GENERIC_FSK software may load T1_CMP with a future EVENT_TMR count value. When EVENT_TMR reaches the T1_CMP value, either (or both) of the following events can be triggered:

1. T1_IRQ (Interrupt)
2. Sequence Command Trigger (can automatically launch or stop a transceiver operation)

Similarly, GENERIC_FSK software may load T2_CMP with a future EVENT_TMR count value. When EVENT_TMR reaches the T2_CMP value, either (or both) of the following events can be triggered:

1. T2_IRQ(Interrupt)

2. Sequence Command Trigger (can automatically launch or stop a transceiver operation)

For either type of T1-triggered events to be enabled, i.e., interrupt or sequence-launch, set T1_CMP_EN=1 after loading T1_CMP with the desired value. If T1_CMP_EN=1, the T1_IRQ interrupt status bit will always become set on a T1 timer match. (Whether or not T1_IRQ generates an GENERIC_FSK Interrupt to the MCU depends on the settings of T1_IRQ_EN and GENERIC_FSK_IRQ_EN, see section: Interrupts). If T1_CMP_EN=0, neither an interrupt nor a sequence command can be triggered by a T1 match.

For either T2-triggered events to be enabled, i.e., interrupt or sequence launch, set T2_CMP_EN=1 after loading T2_CMP with the desired value. If T2_CMP_EN=1, the T2_IRQ interrupt status bit will always become set on a T2 timer match. (Whether or not T2_IRQ generates a GENERIC_FSK Interrupt to the MCU depends on the settings of T2_IRQ_EN and GENERIC_FSK_IRQ_EN, see Section: Interrupts). If T2_CMP_EN=0, neither an interrupt nor a sequence command can be triggered by a T2 match.

See Section Interrupts for a description of how the Link Layer Controller manages the T1 and T2 interrupts, and how software can clear them.

See Section Sequence Commands and Status for a description on how to launch and stop sequences on T1 and T2 timer matches.

45.5.2.3.6 Sequence Commands and Status

The GENERIC_FSK Link Layer Controller provides a command set of 12 transceiver commands, which can be used to launch and stop sequences. GENERIC_FSK software can issue sequence commands by writing to the SEQCMD[3:0] field of the XCVR_CTRL register. The SEQCMD field always reads back what was previously written by software, enabling software to quickly recall the last command written. The sequence commands are as listed in the table below:

SEQCMD[3:0]	COMMAND	DESCRIPTION
0x0	NULL	No Action
0x1	TX_START_NOW	TX Start Now
0x2	TX_START_T1	TX Start @ T1 Timer Compare Match (EVENT_TMR = T1_CMP)
0x3	TX_START_T2	TX Start @ T2 Timer Compare Match (EVENT_TMR = T2_CMP)
0x4	TX_CANCEL	TX Cancel -- Cancels pending TX events but do not abort TX-in-progress[1]
0x5	RX_START_NOW	RX Start Now
0x6	RX_START_T1	RX Start @ T1 Timer Compare Match (EVENT_TMR = T1_CMP)
0x7	RX_START_T2	RX Start @ T2 Timer Compare Match (EVENT_TMR = T2_CMP)
0x8	RX_STOP_T1	RX Stop @ T1 Timer Compare Match (EVENT_TMR = T1_CMP)

Table continues on the next page...

SEQCMD[3:0]	COMMAND	DESCRIPTION
0x9	RX_STOP_T2	RX Stop @ T2 Timer Compare Match (EVENT_TMR = T2_CMP)
0xA	RX_CANCEL	RX Cancel -- Cancels pending RX events but do not abort RX-in-progress[2]
0xB	ABORT	Abort All - Cancels all pending events and abort any sequence-in-progress

1. TX-in-progress implies TX Warmup is complete, and packet transmission is underway (i.e., command decoder state = IN_TX)
2. RX-in-progress implies RX Warmup is complete, and Network Address search or packet reception is underway (i.e., command decoder state = IN_RX)

As described above, sequences can be launched immediately (actually on the next 1MHz clock edge) by issuing one of the following commands:

- TX_START_NOW
- RX_START_NOW

Alternatively, sequences can be scheduled to launch (or stop) in the future on a T1 timer compare, by loading T1_CMP with the desired launch time (relative to the EVENT_TMR), setting T1_CMP_EN=1, and then issuing one of the following commands:

- TX_START_T1
- RX_START_T1
- RX_STOP_T1

Similarly, sequences can be scheduled to launch (or stop) in the future on a T2 timer compare, by loading T2_CMP with the desired launch time (relative to the EVENT_TMR), setting T2_CMP_EN=1, and then issuing one of the following commands:

- TX_START_T2
- RX_START_T2
- RX_STOP_T2

Any transceiver sequence that has been launched, either immediately (by way of the *_NOW commands), or in the future (by way of the *_T1 or *_T2 commands), can be terminated immediately by issuing the ABORT command. This is true whether or not the sequence has actually begun, or is merely pending because the T1- (or T2-) match has not yet occurred. In either case, this is called a software abort. The ABORT command will terminate any active sequence and cancel any pending events.

Regarding the issuance of sequence commands, both immediate and timer-triggered, GENERIC_FSK software should be mindful of the following Guidelines and Restrictions:

PROGRAMING GUIDELINES AND RESTRICTIONS	
1	T1 can only be assigned to 1 function at a time {TX_START, RX_START, RX_STOP}. Once assigned to a function, attempts to change the assignment will be ignored
2	T2 can only be assigned to 1 function at a time {TX_START, RX_START, RX_STOP}. Once assigned to a function, attempts to change the assignment will be ignored
3	To change a T1- or T2- assignment, do one of the following first: **a.** issue an ABORT command **b.** issue an TX_CANCEL command if T1 (or T2) is already assigned to TX_START, and the sequence has not yet started **c.** issue an RX_CANCEL command if T1 (or T2) is already assigned to RX_START or RX_STOP, and the sequence has not yet started
4	T1 and T2 should not be assigned to the same function (results are indeterminate)
5	If T1 and T2 are both assigned to functions, then T1_CMP shall not be equal to T2_CMP (results are indeterminate)
6	If any TX or RX sequence is underway (e.g., TSM not idle), the ongoing sequence will not be interrupted or perturbed by any of the following: **a.** a T1 timer-match with T1 assigned to a function. The sequence trigger will be ignored, but the associated "pending" status bit will be cleared. **b.** a T2 timer-match with T2 assigned to a function. The sequence trigger will be ignored, but the associated "pending" status bit will be cleared. **c.** a TX_START_NOW or RX_START_NOW command. Both commands will be ignored

To perform operations on an RF channel, the GENERIC_FSK Link Layer Controller interacts with the Transceiver Sequence Manager (TSM), which is responsible for the low-level signaling required to activate the individual RF blocks. To do this, the Link Layer Controller includes a Command Decoder. The Command Decoder translates sequence commands (e.g., TX_START_NOW, RX_STOP_T1) into TSM inputs required to start and stop sequences. Then, the TSM executes the warmup, TX, RX, and warmdown functions on the channel. The GENERIC_FSK Command Decoder sends 2 signals (mutually exclusive) to the TSM:

- generic_fsk_tx_en
- generic_fsk_rx_en

Upon receiving an assertion on either of these signals, TSM will launch into warmup on either a TX or RX sequence. When the TSM completes its warmup, it asserts the signal **tsm_wu_complete** back to the Command Decoder, which uses that signal to transition into full-on packet transmit or packet receive. When the packet transmission or reception is complete, the Command Decoder will deassert **generic_fsk_tx_en** (or **generic_fsk_rx_en**), and wait for the TSM to assert **tsm_idle**. At that point, the Command Decoder will return to its IDLE state.

To implement these functions, the Command Decoder includes a small (6-state) finite state machine. The state diagram is shown below, including the TSM interactions.

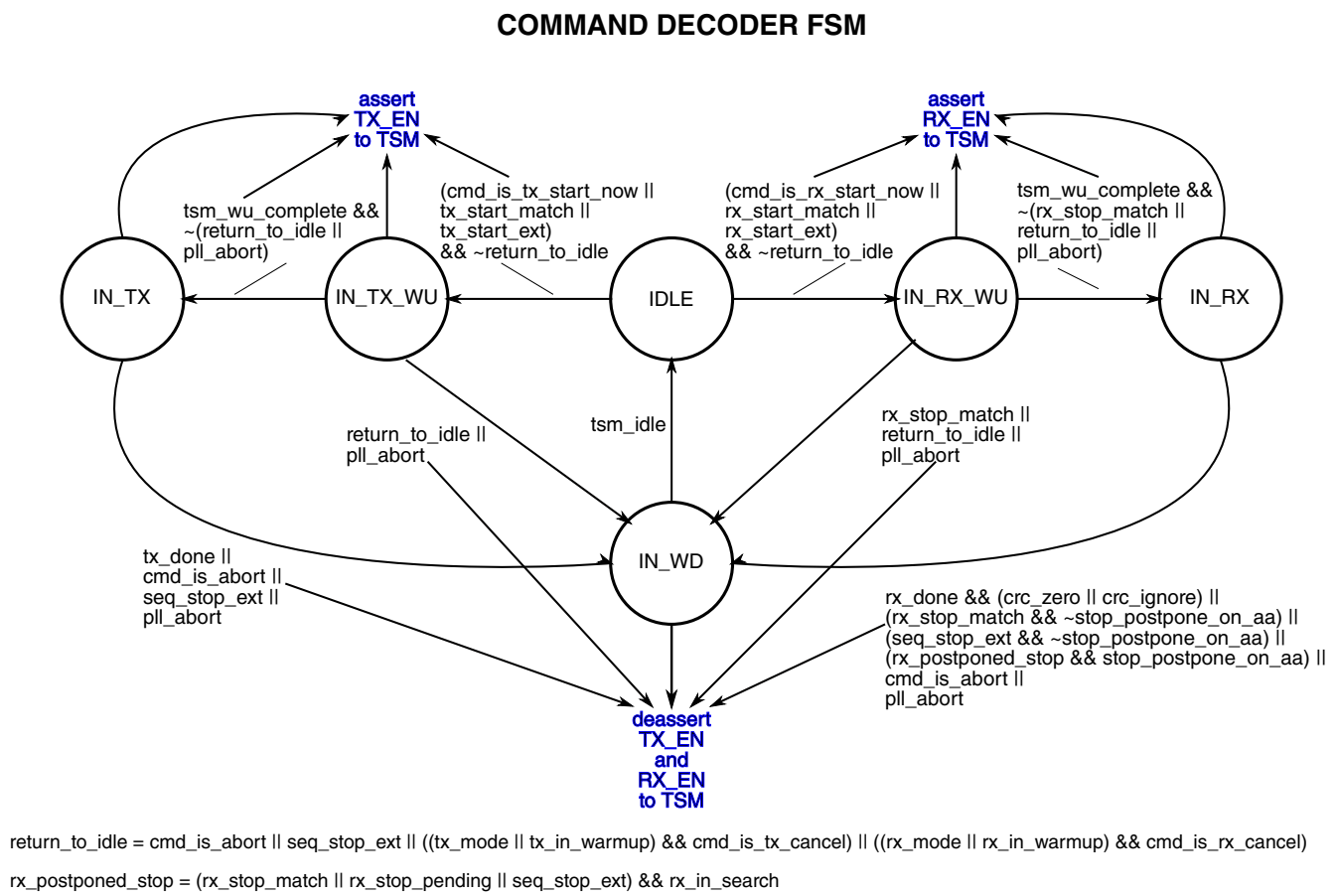


Figure 45-127. GENERIC_FSK COMMAND DECODER FSM

The state of the Command Decoder is carried by the state vector CMDDEC_CS[2:0]. The state can be monitored at all times by reading the CMDDEC_CS field of the XCVR_STATUS register, or via DTEST. The mapping of states to state vector is shown in the table below.

CMDDEC_CS[2:0]	STATE NAME
0b000	IDLE
0b001	IN_TX_WU
0b011	IN_TX
0b101	IN_RX_WU
0b111	IN_RX
0b010	IN_WD

The Command Decoder also maintains an set of 13 status bits, that make it easy for Link Layer software to track, not only state, but pending events; that is, timer-linked events scheduled for future triggering. These status bits reside in the XCVR_STS register. A description of these bits is provided in the table below.

SEQSTS[12:0]	BIT	DESCRIPTION
TX_START_T1_PEND	[0]	TX Sequence will start @ next T1 Match
TX_START_T2_PEND	[1]	TX Sequence will start @ next T2 Match
TX_IN_WARMUP	[2]	TX Sequence in TSM Warmup
TX_IN_PROGRESS	[3]	TX Packet Transmission Currently Underway
TX_IN_WARMDN	[4]	TX Sequence in TSM Warmdown
RX_START_T1_PEND	[5]	RX Sequence will start @ next T1 Match
RX_START_T2_PEND	[6]	RX Sequence will start @ next T2 Match
RX_STOP_T1_PEND	[7]	RX Sequence will stop @ next T1 Match
RX_STOP_T2_PEND	[8]	RX Sequence will stop @ next T2 Match
RX_IN_WARMUP	[9]	RX Sequence in TSM Warmup
RX_IN_SEARCH	[10]	RX Sequence in Network Address Search
RX_IN_PROGRESS	[11]	RX Packet Reception Currently Underway
RX_IN_WARMDN	[12]	RX Sequence in TSM Warmdown

45.5.2.3.7 Interrupts

Interrupt Status Bit Structure

The Generic FSK Link Layer Controller has 10 interrupt sources, each represented in the register map by an interrupt status bit. All interrupt status bits share a common, generic structure. If a particular interrupt source is enabled, a “TRIGGERING EVENT” for that interrupt source, will always set the status bit. A write-1-to-clear input from the IPS bus, will clear the status bit, but only if there is not a simultaneous triggering event. The triggering event prevails over a software clear attempt, if both events coincide. An integrated clock gate guarantees that the status bit only sees a clock when either a triggering event occurs, or a write-1-to-clear pulse arrives from the IPS bus. This reduces interrupt status bit power consumption to an absolute minimum. (Note: the TSM_IRQ does not have an associated status bit in the Link Layer Controller; its status bit resides in the Transceiver Sequence Manager, or TSM).

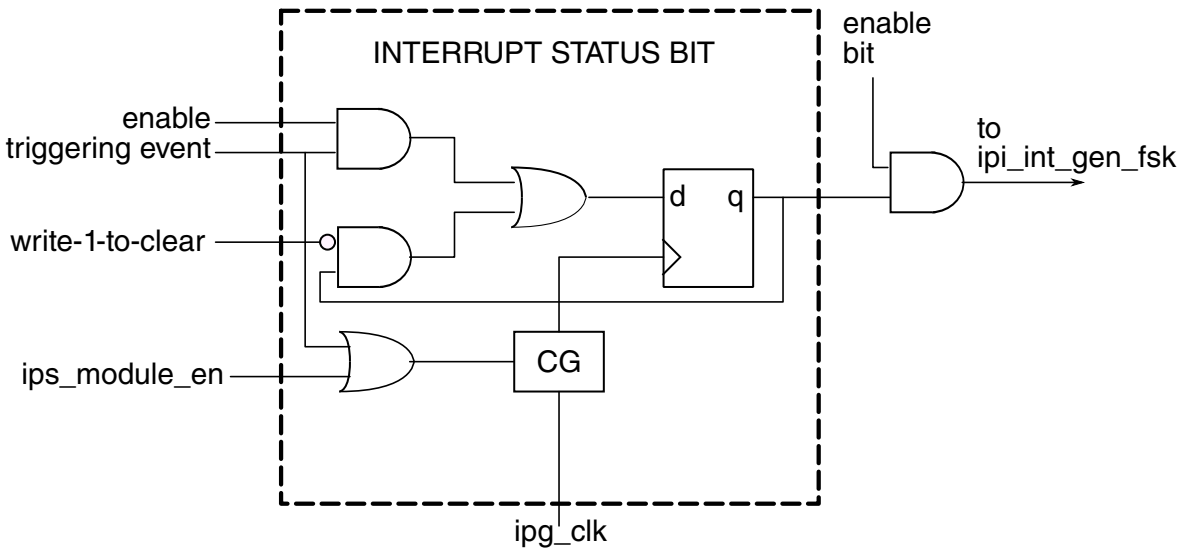


Figure 45-128. Interrupt Status Bit

Each GENERIC_FSK interrupt status bit, has an ENABLE bit associated with it. The name of the enable bit tracks the name of the interrupt source, with the string "_EN" appended at the end. Note that for any GENERIC_FSK interrupt source, if the triggering event occurs, the Interrupt Status Bit will be set regardless of the state of the corresponding ENABLE bit. If any of the 10 interrupts is to be ignored, software should clear the corresponding ENABLE bit, and apply the appropriate bit mask when reading the IRQ_CTRL register, to mask out the unwanted status bit.

GENERIC_FSK Interrupt Architecture

The 10 interrupt sources (status bits) are combined with their individual ENABLE bits, and then logically OR'ed together, in sum-of-products fashion, to generate single interrupt line to the MCU: **ipi_int_gen_fsk**. A global mask bit, GENERIC_FSK_IRQ_EN, can enable or disable **ipi_int_gen_fsk** altogether. There is no prioritization of interrupt sources; they have equal weight. The **ipi_int_gen_fsk** output is a level-sensitive indicator and will remain asserted until all interrupt status bits are cleared (or the corresponding ENABLE bits cleared). The following diagram depicts the GENERIC_FSK Interrupt Architecture.

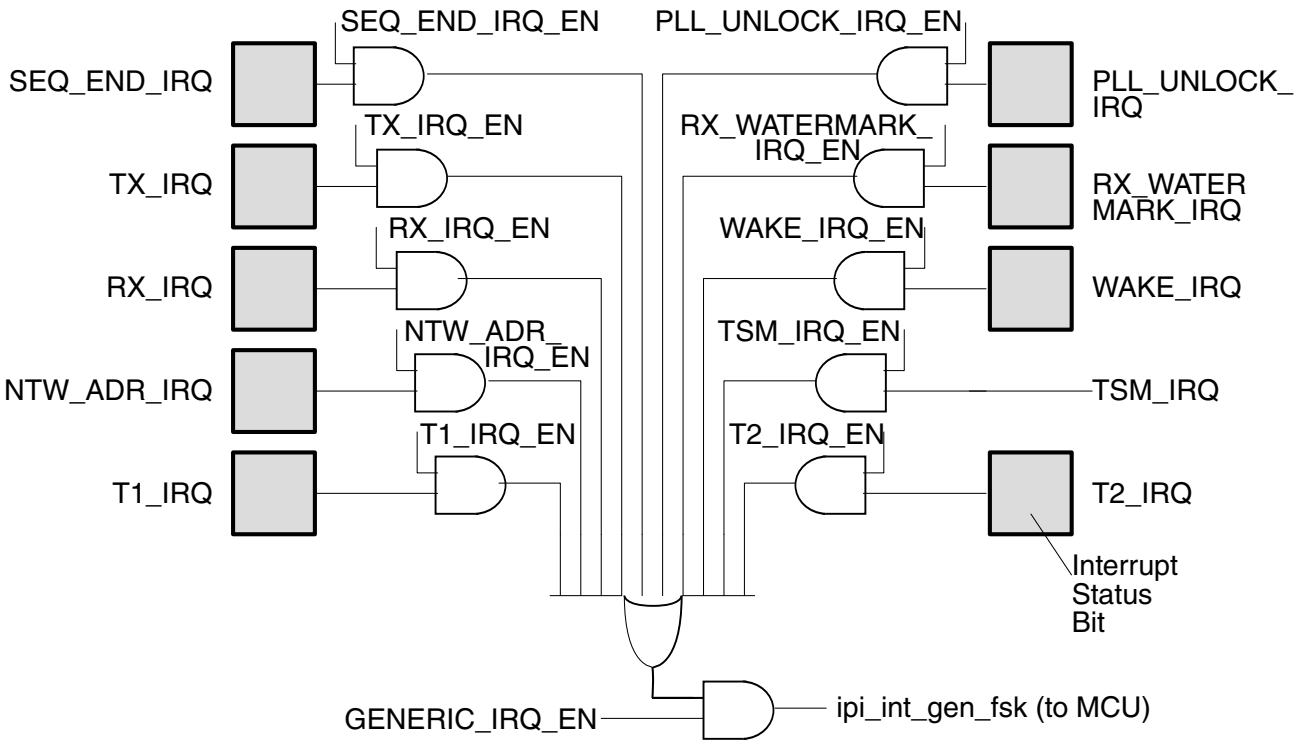


Figure 45-129. GENERIC_FSK Interrupt Architecture

Clearing Interrupts

All interrupt status bit use a write-1-to-clear protocol. Writing a '1' to the interrupt status bit, in the IRQ_CTRL register, clears the offending interrupt. Writing a '0' to an interrupt status bit has no effect on the bit. Interrupt status bits are not affected by reads.

GENERIC FSK Interrupt Sources

The following table describes the 10 GENERIC_FSK Interrupt Sources

Interrupt	Description
T1_IRQ, T2_IRQ	The GENERIC_FSK Link Layer features a 24-bit Event Timer, which runs at a rate of 1MHz. The Link Layer has 2 Timer interrupts (T1_IRQ, T2_IRQ), each with its own 24-bit compare register (T1_CMP, T2_CMP), and each with its own compare enable (T1_CMP_EN, T2_CMP_EN). For each timer compare enable, if the bit is set, a match on the respective 24-bit compare value to the Event Timer will cause the corresponding interrupt status bit to become set. If the compare enable is not set, Event Timer matches won't cause the corresponding interrupt status bit to become set.
TX_IRQ	TX interrupt occurs at the end of a TX operation.
RX_IRQ	RX interrupt occurs at the end of a successful RX packet reception. A successful packet reception implies CRC was verified to be good, H0 matching passed, H1 matching passed, and received packet length did not violate pre-set

Table continues on the next page...

Interrupt	Description
	limits. Nominally, RX interrupts are not generated on packets which fail CRC check, or which failure the H0, H1 or LENGTH limits. A received packet with a failed CRC check or a H0, H1, or LENGTH violation, results in an RX recycle back to the Network Address Search state. To receive a Data Indication (RX_IRQ) on packets which fail CRC, set the bit CRC_IGNORE=1. H0 and H1 matching, and LENGTH limiting, are under SW control.
NTW_ADR_IRQ	A Network Address Match has occurred on an enabled Network Address. At NTW_ADR_IRQ, a timestamp is captured by transferring the current contents of the EVENT_TMR into the TIMESTAMP register. Software can determined which Network Address matched by querying the NTW_ADR_MCH[3:0] field of the NTW_ADR_CTRL register.
SEQ_END_IRQ	The Sequence End Interrupt (SEQ_END_IRQ), indicates that a transceiver operation (TX or RX) has completed, and the Command Decoder State Machine, and the Transceiver Sequence Manager (TSM), have returned to their respective IDLE states. A SEQ_END_IRQ will always occur at the end of a sequence, even if the sequence terminated abnormally (such as a Software Abort, or a PLL Unlock Abort). A SEQ_END_IRQ always occurs whenever the Command Decoder FSM transitions from any non-idle to IDLE state. When SEQ_END_IRQ occurs, software can be sure that the Command Decoder and TSM are in their idle state, and a new sequence can be programmed.
RX_WATERMARK_IRQ	RX Watermark interrupt will occur during packet reception, when the number of received bytes, as indicated by the read-only BYTE_COUNTER register, matches the contents of the RX_WATERMARK register. For the purpose of defining RX_WTR_MARK, the first byte received (BYTE_COUNTER = 0) is the first byte of Network Address; the second byte received (BYTE_COUNTER = 1) is the second byte of Network Address, etc. To cause an RX Watermark Interrupt to occur after a 2-byte Network Address has been received, set RX_WTR_MARK=1. By default, RX_WATERMARK is set to a value larger than the longest supported GENERIC_FSK packet length, so an RX_WATERMARK_IRQ won't be triggered under these conditions.
PLL_UNLOCK_IRQ	When an PLL unlock event occurs during an transceiver operation, and the transceiver has been configured to automatically abort sequences on PLL unlock events, the PLL_UNLOCK_IRQ status bit will become set. The Transceiver Sequence Manager (TSM) will begin monitoring for PLL unlock only after the PLL has been given sufficient time to achieve lock; A PLL unlock which occurs after the warmup period, can be enabled to cause a sequence abort. Individual enables are provided for TX and RX sequences. (See the TSM Chapter for more details).
WAKE_IRQ	A WAKE_IRQ will be triggered when the GENERIC_FSK Link Layer Controller has awoken from a DSM (Deep Sleep Mode) cycle. WAKE_IRQ indicates that the RF Oscillator has been restarted, and the GENERIC_FSK EVENT_TMR has resumed counting.

Table continues on the next page...

Interrupt	Description
TSM_IRQ	TSM_IRQ is a debug feature, enabling the Transceiver Sequence Manager (TSM) to generate an interrupt at any point in a TX or RX Warmup. TSM is a multipurpose hardware resource shared by all of the protocol engines in the SoC. Thus, TSM does not have its own interrupts; its interrupts are assigned to whichever link layer controller is currently executing an RF operation. TSM Interrupt Status bits reside in Transceiver (XCVR) Address Space, and can be cleared there. There is no intended mission-mode use for TSM_IRQ. See the TSM Chapter for more details.

45.5.2.3.8 Memory Mapping

In the SoC memory map, space has been allocated to GENERIC_FSK. Within the SoC, peripherals are allocated memory space in 4KB blocks, called IPS slots. GENERIC_FSK occupies one such slot. GENERIC_FSK address space consists of the GENERIC_FSK registers. The partitioning of address space for GENERIC_FSK is shown in the diagram below.

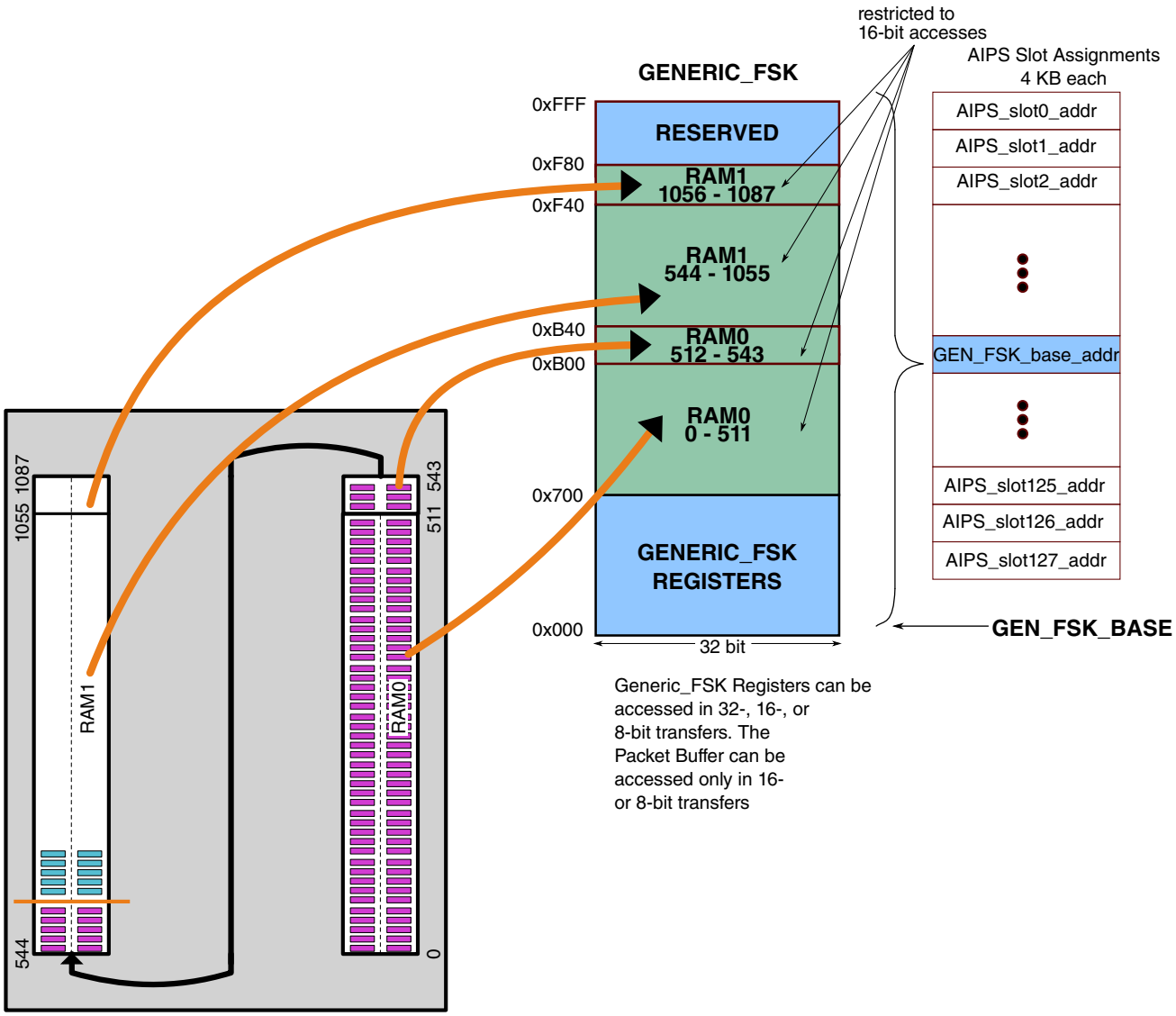


Figure 45-130. GENERIC_FSK MEMORY MAPPING

NOTE

See AIPS Memory Map for specific memory assignments details.

All registers within GENERIC_FSK memory space, are accessible by the MCU in 8-, 16-, or 32-bit accesses. For efficiency, 32-bit accesses are recommended.

45.5.2.3.9 Sequence Timing Diagrams

The following timing diagram depicts an example of GENERIC_FSK packet transmission, highlighting the following events:

1. Software stores packet octets into Packet Buffer, to be transmitted

2. Software loads T1_CMP with desired TX start time, sets T1_CMP_EN=1
3. Software schedules TX sequence with TX_START_T1 command
4. EVENT_TMR matches T1_CMP, TX sequence starts automatically
5. Generic_FSK Link Layer reads octets from Packet Buffer. Packet is transmitted
6. TX_IRQ asserts after last bit of packet transmitted
7. SEQ_END_IRQ asserts after TSM returns to IDLE (TX warmdown complete)

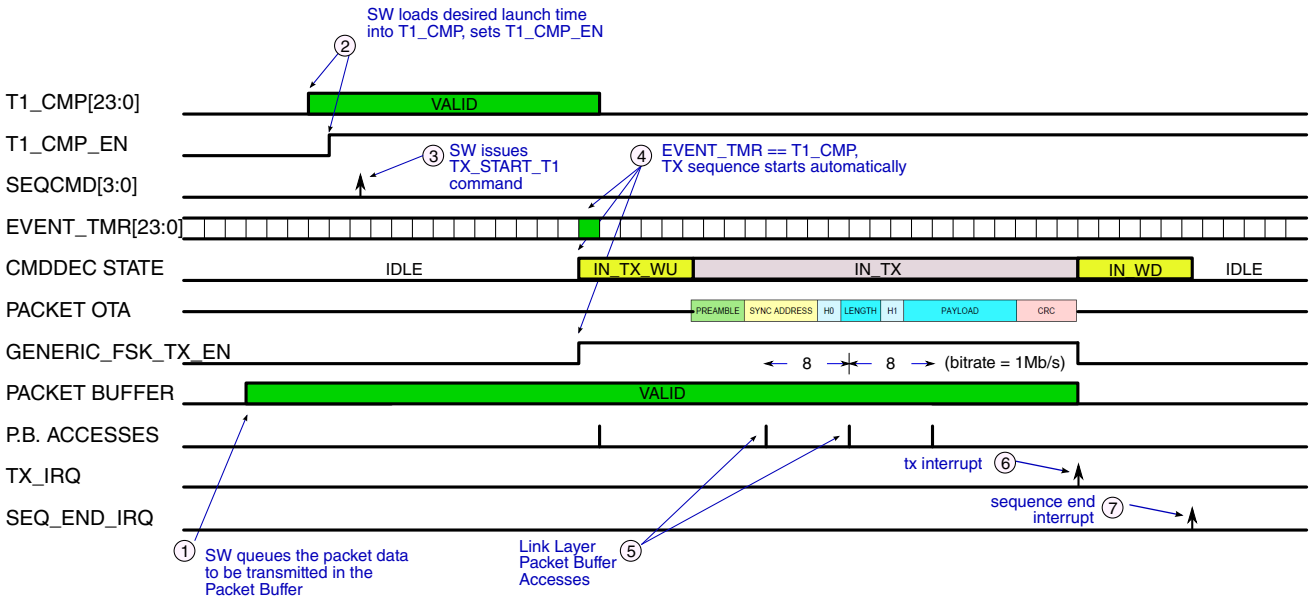


Figure 45-131. GENERIC FSK TX SEQUENCE: LAUNCH VIA T1 TIMER

The following timing diagram depicts an example of GENERIC_FSK “good” packet reception, highlighting the following events:

1. Software loads T2_CMP with desired RX start time, sets T2_CMP_EN=1
2. Software schedules RX sequence with RX_START_T2 command
3. EVENT_TMR matches T2_CMP, RX sequence starts automatically
4. Network Address search, followed by NTW_ADR_IRQ interrupt, TIMESTAMP capture
5. Packet reception, Generic_FSK LL write octets to Packet Buffer
6. CRC verification passes, and no H0, H1, or LENGTH_MAX violations, RX_IRQ asserted, Packet RX Buffer valid, RSSI valid
7. SEQ_END_IRQ asserts after TSM returns to IDLE (RX warmdown complete)

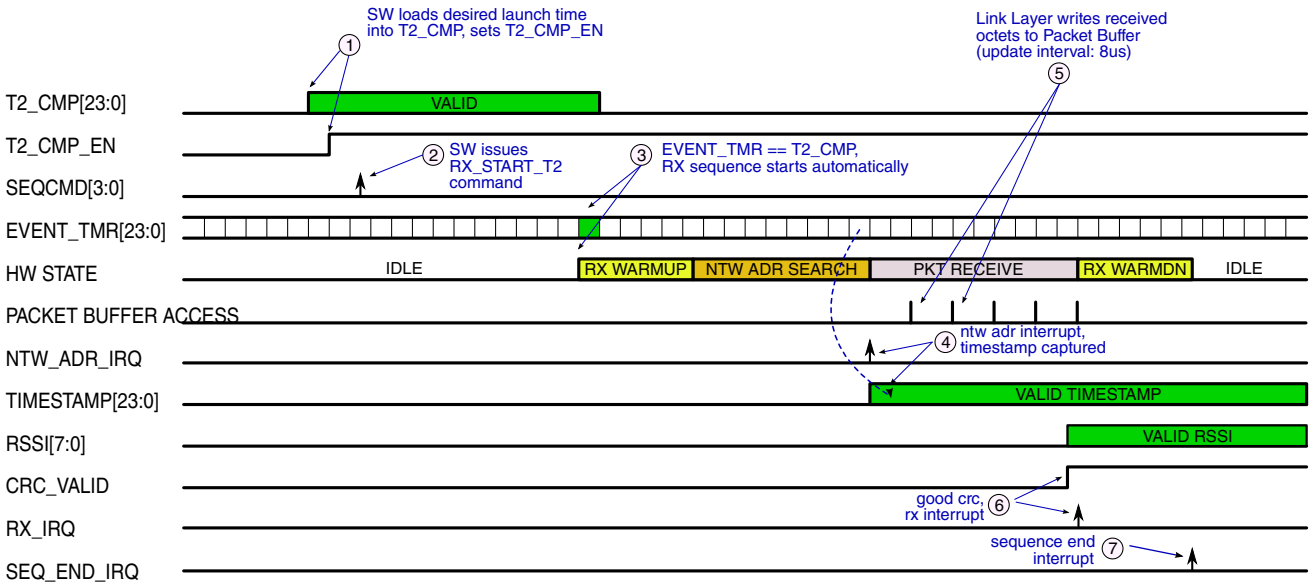


Figure 45-132. GENERIC FSK RX SEQUENCE: LAUNCH VIA T2 TIMER, GOOD PACKET

The following timing diagram depicts an example of GENERIC_FSK “bad” packet reception, highlighting the following events:

1. Software loads T2_CMP with desired RX start time, sets T2_CMP_EN=1
2. Software schedules RX sequence with RX_START_T2 command
3. EVENT_TMR matches T2_CMP, RX sequence starts automatically
4. Network Address search, followed by NTW_ADR_IRQ interrupt, TIMESTAMP capture
5. Packet reception, Generic_FSK LL write octets to Packet Buffer
6. CRC verification fails, no CRC_VALID, no RX_IRQ, RX Recycle
7. No SEQ_END_IRQ, return to Network Address Search

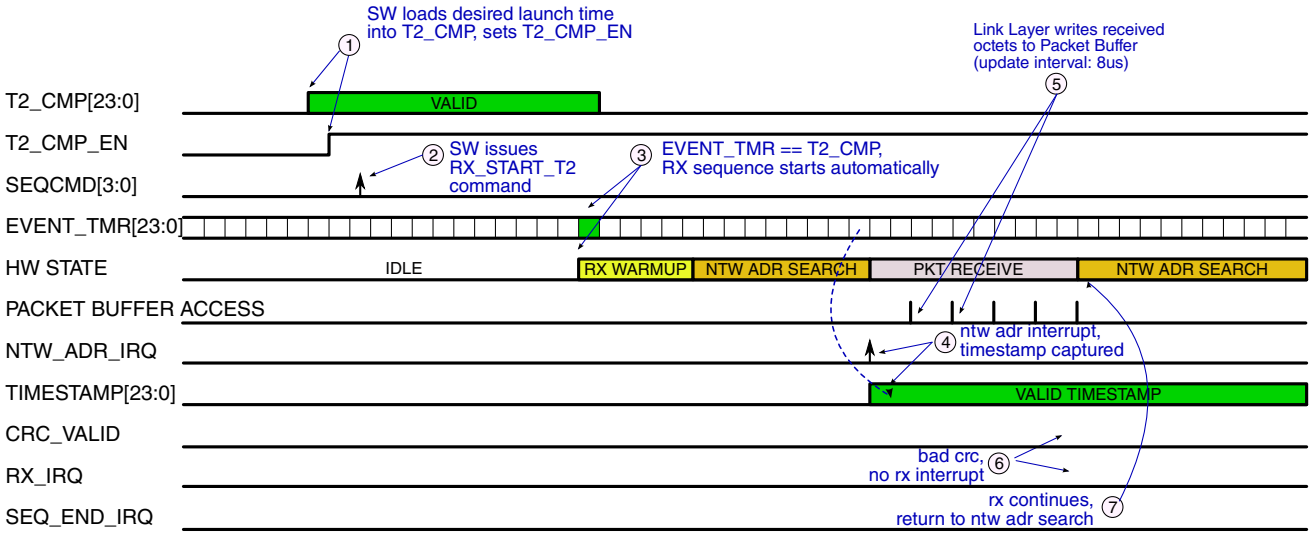


Figure 45-133. GENERIC FSK RX SEQUENCE: LAUNCH VIA T2 TIMER, BAD PACKET, RX RECYCLE

The following timing diagram depicts a GENERIC_FSK RX search operation, which fails to detect a Network Address before timing out. The following events are highlighted:

1. Software loads T1_CMP with desired RX start time, sets T1_CMP_EN=1
2. Software loads T2_CMP with desired RX stop time, sets T2_CMP_EN=1
3. Software schedules RX sequence start with RX_START_T1 command
4. Software schedules RX sequence end with RX_START_T2 command
5. EVENT_TMR matches T1_CMP, RX sequence starts automatically
6. EVENT_TMR matches T2_CMP, RX sequence goes into warmdown automatically
7. SEQ_END_IRQ asserts to indicate TSM has returned to IDLE. No NTW_ADR_IRQ, No RX_IRQ.

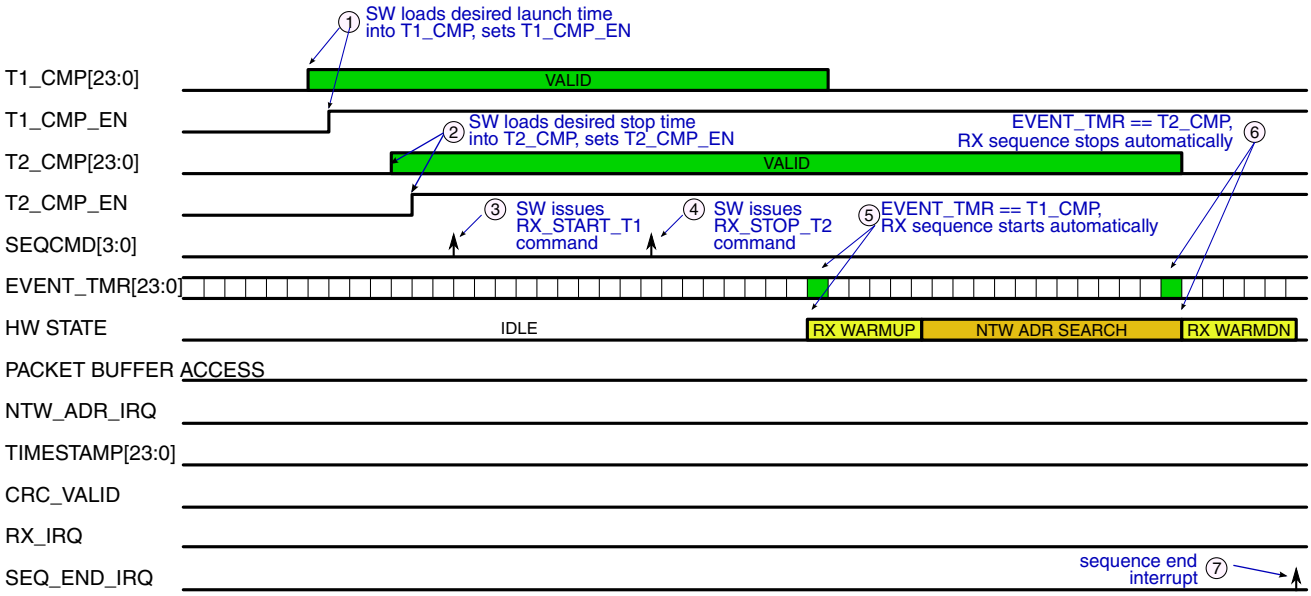


Figure 45-134. GENERIC FSK RX SEQUENCE: SLAVE CHANNEL SEARCH, LAUNCH VIA T1, STOP VIA T2

45.5.2.3.10 Deep Sleep Mode

When the GENERIC_FSK Link Layer anticipates long periods of inactivity, the Link Layer controller can be put into a Deep Sleep Mode (DSM), where all of its clocks are turned off. In addition, the GENERIC_FSK Link Layer can be configured to optionally turn off the RF Oscillator, if it is not otherwise needed by the SoC. Deep Sleep Mode results in dramatic power savings.

To make possible Deep Sleep Mode, the transceiver incorporates a low-frequency, high-precision Deep Sleep Timer, DSM_TIMER. This timer is clocked by a crystal-referenced 32.768KHz oscillator. The timer is 24-bit wide, yielding a 8.5 minute rollover (the maximum length of a deep sleep period). The DSM_TIMER resides in the RSIM module.

The hardware to implement DSM within the SoC, is partitioned across several modules, and several power domains. This is to allow the maximum achievable power savings, by retaining full voltage for only those modules which need to be active in DSM, and placing the remaining modules in “state retention” mode. The diagram below depicts the partitioning of DSM hardware across the various modules, including the GENERIC_FSK Link Layer Controller.

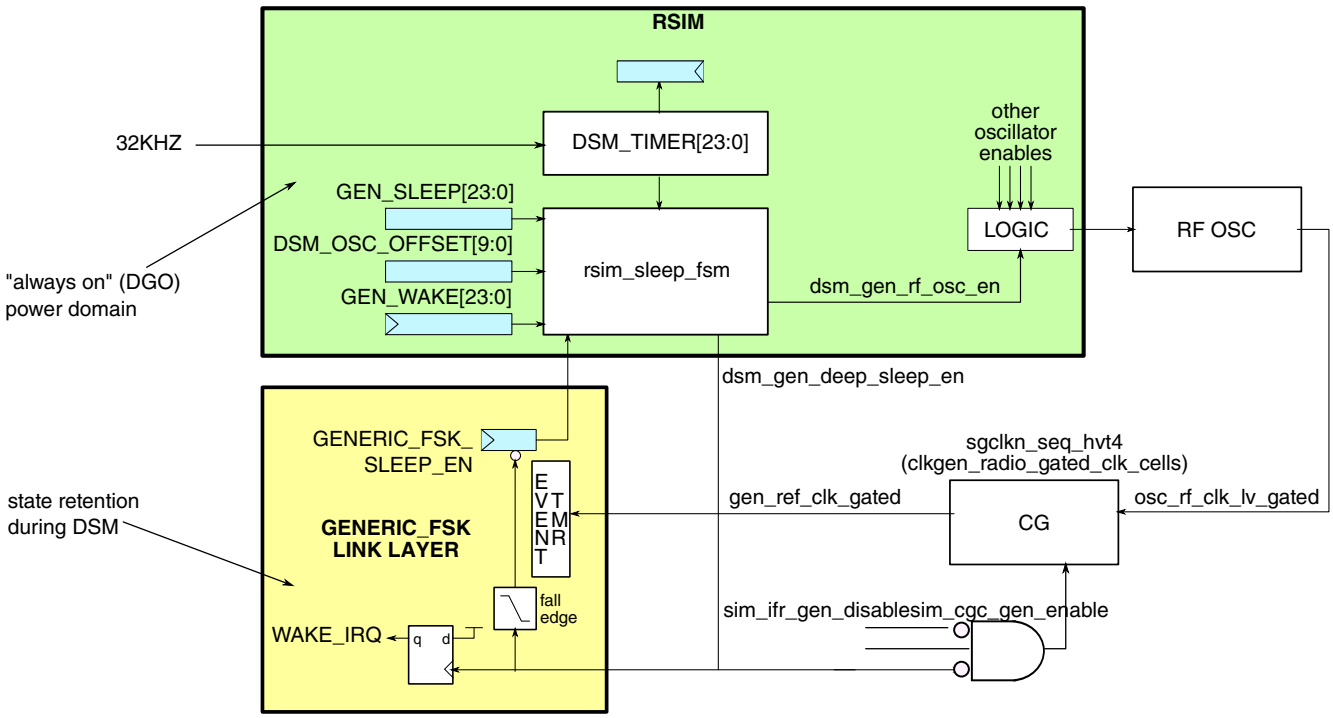


Figure 45-135. Deep Sleep Mode

The concept behind DSM, as it is implemented for GENERIC_FSK, is to allow the GENERIC_FSK timebase to be maintained for a period of time, while the RF Oscillator is turned off and much of the SoC is placed in state-retention. During DSM, the timebase management is temporarily transferred from the GENERIC_FSK EVENT_TMR to the low-power DSM_TIMER, and after DSM exit, timebase management is transferred back to the EVENT_TMR. During DSM, the EVENT_TMR is frozen. The total time spent in DSM is known to software, so that upon exiting DSM, the EVENT_TMR can be updated to correct for the precise amount of time it was frozen.

DSM entry and exit, and RF Oscillator re-start during DSM, are governed by 5 registers, as described in the table below.

Field	R/W	Description
GEN_SLEEP[23:0]	rw	If DSM_CTRL[GEN_SLEEP_REQUEST]=1, enter DSM and freeze EVENT_TMR when GEN_SLEEP matches DSM_TIMER. **NOTE:** This register resides in RSIM Address Space
DSM_OSC_OFFSET[9:0]	rw	Awaken SoC, start the RF Oscillator when: (GEN_WAKE - DSM_OSC_OFFSET) = DSM_TIMER **NOTE:** This register resides in RSIM Address Space
GEN_WAKE[23:0]	rw	Exit Deep Sleep Mode, and resume EVENT_TMR, when GEN_WAKE matches DSM_TIMER **NOTE:** This register resides in RSIM Address Space

Table continues on the next page...

Field	R/W	Description
GEN_SLEEP_REQUEST	w	Enable a match on GEN_SLEEP[23:0] to DSM_TIMER[23:0], to enter Deep Sleep Mode, by writing a 1 to this bit. This bit is write-only, and always reads back 0. Writing a 0 to this bit has no effect. This bit resides in the DSM_CTRL register of GENERIC_FSK address space.
DSM_TIMER[23:0]	r	Current State of the 32KHz Sleep Timer **NOTE:** This register resides in RSIM Address Space

Procedure

The procedure to schedule a DSM cycle (entry/exit) as well as to program the desired RF Oscillator re-start time, and the calculation to restore the GENERIC_FSK EVENT_TMR after a DSM cycle, is described below.

1. Software determines the future time at which it would like to enter DSM, by reading the DSM_TIMER, computing the number of 32KHz clock cycles remaining until the desired DSM start-time, and writing this value into GEN_SLEEP register. The value programmed into GEN_SLEEP should be no fewer than 4 clocks greater (in the future) than the current time as read from DSM_TIMER.
2. Software determines the future time at which it would like to exit DSM, by computing the number of 32KHz clock cycles remaining until the desired DSM exit-time, and writing this value into GEN_WAKE register.
3. The DSM_OSC_OFFSET register should be pre-programmed with the number of 32KHz required for oscillator re-start. This should be done during initial SoC configuration and should not need to be updated during this procedure. **Note:** When programming GEN_WAKE, software must ensure that the following 2 equations are true:

   GEN_WAKE – DSM_OSC_OFFSET > DSM_TIMER + 4

   GEN_WAKE – DSM_OSC_OFFSET > GEN_SLEEP

4. Software writes a 1 to DSM_CTRL[GEN_SLEEP_REQUEST]. This bit resides in GENERIC_FSK address space. This bit is write-only and always reads back 0. Writing this bit to 1 enables a DSM cycle to commence using the values programmed into GEN_SLEEP and GEN_WAKE.
5. MCU can go into a low power state (e.g., wait-for-interrupt to enter STOP)
6. When DSM_TIMER = GEN_SLEEP, the EVENT_TMR will be clock-gated so that it will remain frozen at its current count. The RF Oscillator will be turned off (unless overridden due to a non-GENERIC_FSK-related SoC requirement)
7. All hardware not in the low-voltage domain will be placed into state-retention (or other low-power state).
8. When DSM_TIMER = (GEN_WAKE – DSM_OSC_OFFSET), the parts of the SoC in low-power state will be returned to normal operation, and the RF Oscillator will be

re-started. (DSM_OSC_OFFSET will be programmed with a sufficient value to allow the oscillator to stabilize before clocks are released to the GENERIC_FSK Link Layer)

9. When DSM_TIMER = GEN_WAKE, clocking of the GENERIC_FSK Link Layer Controller will resume and the EVENT_TMR will be ungated (allowed to resume counting).
10. When DSM_TIMER = GEN_WAKE, the WAKE_IRQ status bit of IRQ_CTRL bit will be asserted, and will generate an interrupt to the MCU if IRQ_CTRL[WAKE_IRQ_EN]=1. Clear WAKE_IRQ by writing a '1' to the bit location.
11. Software computes the time that the SoC was in DSM (and hence the time the EVENT_TMR was frozen), in microseconds, with the equation:

    (GEN_WAKE – GEN_SLEEP) / 32768 * 1000000

12. Software increments the EVENT_TMR by this amount, by writing this value to EVENT_TMR[23:0] register with EVENT_TMR_ADD=1.
13. On the next 1MHz clock edge, the EVENT_TMR has been restored to where it would have been, had no DSM occurred, with 0.5-microsecond accuracy.

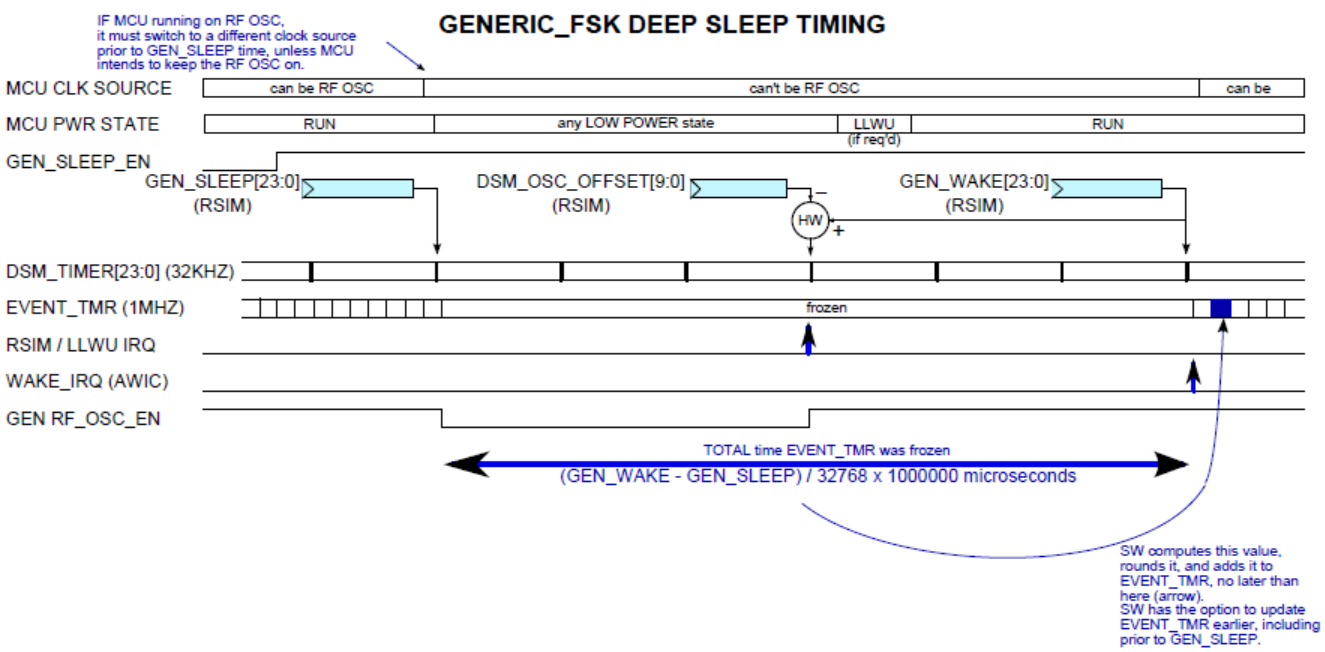


Figure 45-136. GENERIC FSK Deep Sleep Mode Timing

Note: If the DSM low power state selected by the SoC is VLLS, the SIM_CGC bit for GENERIC_FSK will be cleared, and must be re-established in the LLWU ISR (not the WAKE_IRQ ISR, which would be too late)

45.5.2.3.11 Multi-protocol Arbitration

The GENERIC_FSK Link Layer controller may be included in an SoC that includes similar controllers for other protocols (e.g., Bluetooth Low Energy). GENERIC_FSK may be required to operate simultaneously with other protocols on the SoC, and there are scenarios where both protocol engines require access to the RF channel at the same time.

To assist in software arbitration between GENERIC_FSK and other protocols, the read-only status but XCVR_BUSY has been included in the XCVR_STS register. When this bit is set, this indicates that the RF channel is currently in use (TSM is busy), by one of the Radio's four link layer controllers. When this bit it clear, the RF channel is available to GENERIC_FSK (TSM is idle).

45.5.2.3.12 Wifi Coexistence

Provisions have been made to allow for the 2.4GHz transceiver to coexist in the same space as a WiFi transceiver IC, which shares the same frequency band. The coexistence scheme designates the WiFi transceiver as the master, and the 2.4GHz transceiver as the slave. The objective of the coexistence strategy is to prevent both the WiFi and 2.4GHz transceivers transmitting simultaneously; a configuration option exists to also prevent the 2.4GHz transceiver from receiving when the WiFi transceiver owns the RF channel.

A very basic mechanism is used in cases where 2.4GHz radio does not generate requests, but instead defers to the WiFi transceiver to grant permission to use the medium. The WiFi IC generates a signal, RF_NOT_ALLOWED. If this signal is asserted, then 2.4GHz radio does not perform any communication. When this signal is de-asserted, then 2.4GHz radio is free to perform communications. In case the signal RF_NOT_ALLOWED is de-asserted when 2.4GHz radio has already initiated the transmission/reception of a packet then the 2.4GHz radio must stop its activity immediately .

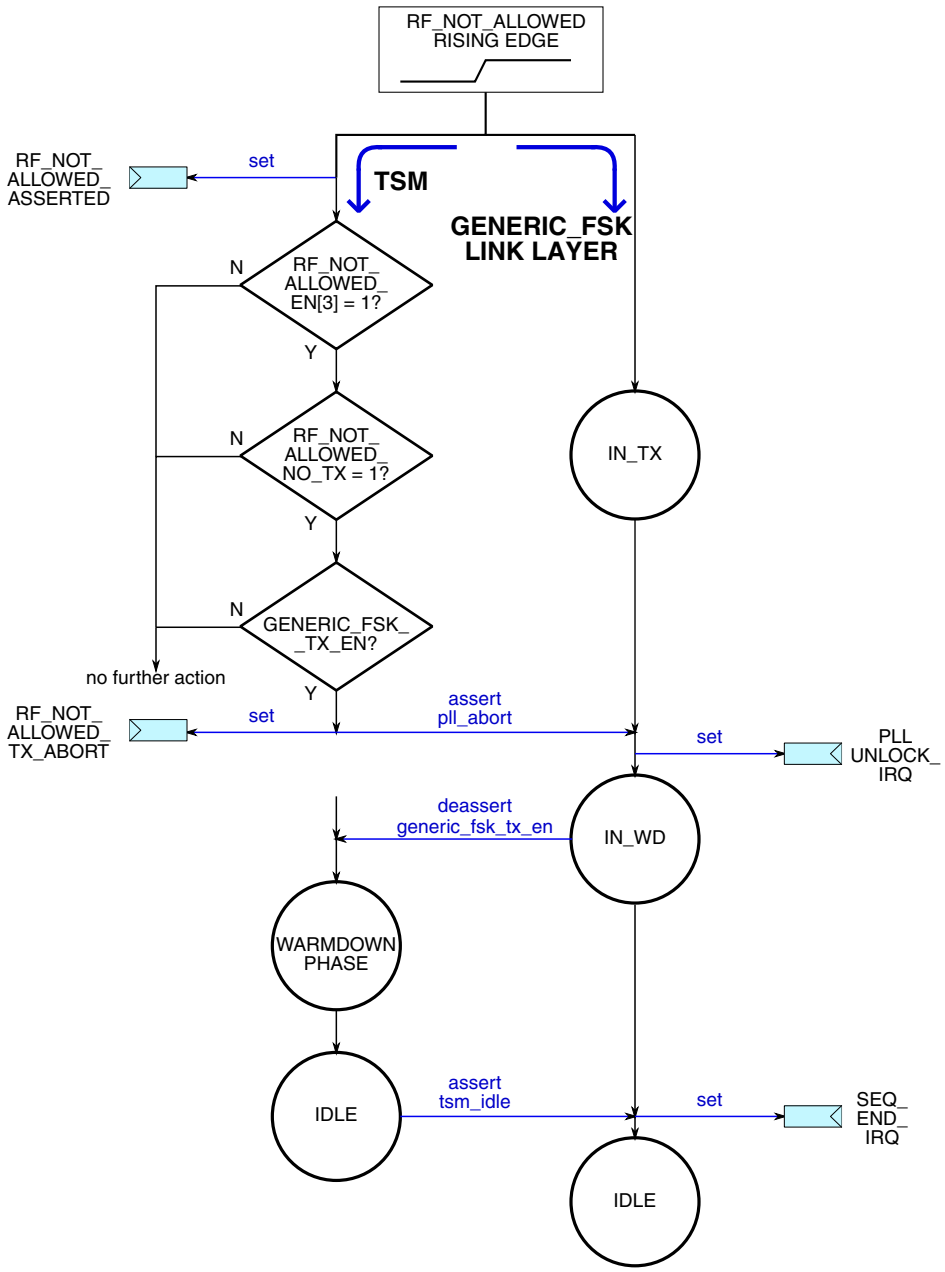
When RF_NOT_ALLOWED aborting is enabled, the GENERIC_FSK Link Layer hardware must respond to RF_NOT_ALLOWED assertions, by aborting any TX or RX sequence which is currently underway, and Link Layer software must not initiate any new sequences until RF_NOT_ALLOWED has been deasserted. The hardware response must be handled autonomously by the Link Layer, with no MCU intervention required.

In the multi-protocol 2.4GHz radio, RF_NOT_ALLOWED aborting can be individually enabled/disabled for each protocol engine. For GENERIC_FSK, RF_NOT_ALLOWED_EN[3] is the associated enable control bit. This bit resides in the COEX_CTRL register in XCVR address space. When this bit is 0, transitions on RF_NOT_ALLOWED are ignored by the GENERIC_FSK Link Layer hardware; when this bit is 1, the GENERIC_FSK Link Layer hardware will monitor RF_NOT_ALLOWED at all times, and abort any active sequence which is underway when an assertion on the pin occurs. The complete hardware response to RF_NOT_ALLOWED assertions is described below.

Additional control over RF_NOT_ALLOWED aborting is provided by the RF_NOT_ALLOWED_NO_TX and RF_NOT_ALLOWED_NO_RX control bits. If RF_NOT_ALLOWED_NO_TX=1, then an RF_NOT_ALLOWED abort will occur only if a GENERIC_FSK TX sequence is underway (generic_fsk_tx_en=1). GENERIC_FSK TX sequences will not be aborted if RF_NOT_ALLOWED_NO_TX=0. If RF_NOT_ALLOWED_NO_RX=1, then an RF_NOT_ALLOWED abort will occur only if GENERIC_FSK RX sequence is underway (generic_fsk_rx_en=1). GENERIC_FSK RX sequences will not be aborted if RF_NOT_ALLOWED_NO_RX=0. The RF_NOT_ALLOWED_NO_TX and RF_NOT_ALLOWED_NO_RX control bits reside in the COEX_CTRL register.

For the purposes of triggering a hardware abort, the *pll_abort* input to the GENERIC_FSK Link Layer hardware is used. This is because the hardware response to the RF_NOT_ALLOWED assertion is identical to that of a PLL unlock event. This also means that, when RF_NOT_ALLOWED aborting is enabled for GENERIC_FSK the PLL_UNLOCK_IRQ interrupt status bit, will be dual purpose: it will not only indicate a PLL unlock condition, but also a RF_NOT_ALLOWED abort. PLL aborting and RF_NOT_ALLOWED aborting are enabled separately, with the COEX_CTRL register maintaining the control bits required for the latter. The status bits for RF_NOT_ALLOWED aborting are also available in COEX_CTRL, so that software will be able to distinguish the source of the PLL_UNLOCK_IRQ, in case both PLL and RF_NOT_ALLOWED aborting are enabled.

The sequence of events which results in a hardware abort of an GENERIC_FSK TX operation triggered by an assertion on RF_NOT_ALLOWED, is a collaboration between the TSM (Transceiver Sequence Manager) and the GENERIC_FSK Link Layer hardware, as shown in the following diagram.



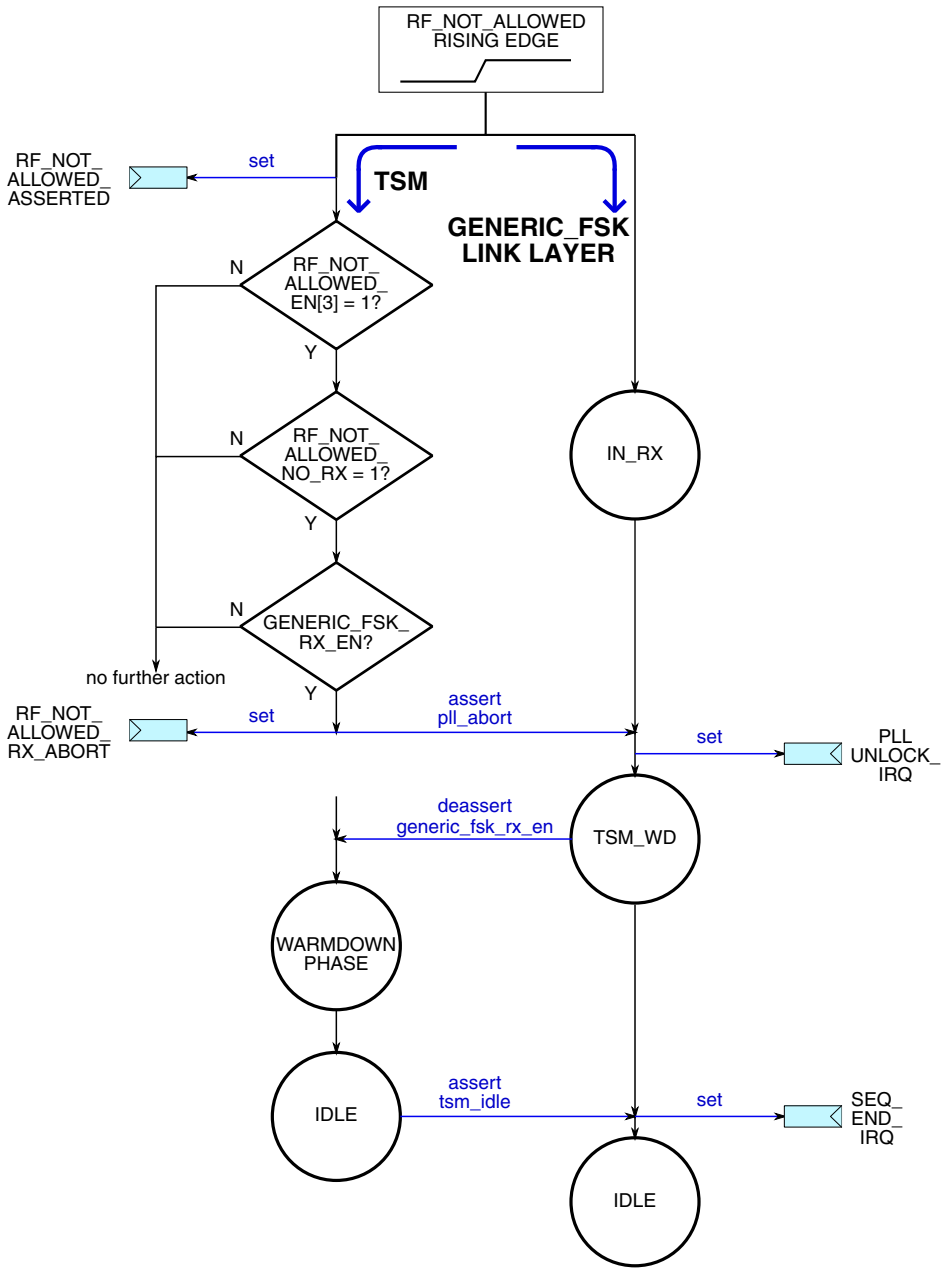
Upon assertion on RF_NOT_ALLOWED, the TSM sets the RF_NOT_ALLOWED_ASSERTED status bit in COEX_CTRL, then checks that the conditions for a hardware abort are all met:

- RF_NOT_ALLOWED_EN[3] = 1, which enables GENERIC_FSK to respond to RF_NOT_ALLOWED events
- RF_NOT_ALLOWED_NO_TX = 1, which enables TX operations to be aborted
- generic_fsk_tx_en = 1, TX request to TSM from GENERIC_FSK, indicating TX operation in progress

If all conditions are not met, no further action is taken. Otherwise, the TSM sets COEX_CTRL[RF_NOT_ALLOWED_TX_ABORT], and also asserts *pll_unlock* to the GENERIC_FSK Link Layer hardware. GENERIC_FSK responds by asserting IRQ_CTRL[PLL_UNLOCK_IRQ] in GENERIC_FSK address space. GENERIC_FSK enters IN_WD state, which deasserts generic_fsk_tx_en to the TSM. This initiates the TSM TX warmdown. GENERIC_FSK holds in IN_WD to wait for TSM to return to idle. Once this occurs, GENERIC_FSK asserts IRQ_CTRL[SEQ_END_IRQ] and returns to its IDLE state. Three status bits are now set to indicate to software that the source of the abort was an RF_NOT_ALLOWED assertion:

- IRQ_CTRL[PLL_UNLOCK_IRQ]
- COEX_CTRL[RF_NOT_ALLOWED_ASSERTED]
- COEX_CTRL[RF_NOT_ALLOWED_TX_ABORT]

The sequence of events which results in a hardware abort of an GENERIC_FSK RX sequence triggered by an assertion on RF_NOT_ALLOWED, is shown in the following diagram.



Upon assertion on RF_NOT_ALLOWED, the TSM sets the RF_NOT_ALLOWED_ASSERTED status bit in COEX_CTRL, then checks that the conditions for a hardware abort are all met:

- RF_NOT_ALLOWED_EN[3] = 1, which enables GENERIC_FSK to respond to RF_NOT_ALLOWED events
- RF_NOT_ALLOWED_NO_RX = 1, which enables RX operations to be aborted
- generic_fsk_rx_en = 1, RX request to TSM from GENERIC_FSK, indicating RX operation in progress

If all conditions are not met, no further action is taken. Otherwise, the TSM sets COEX_CTRL[RF_NOT_ALLOWED_RX_ABORT], and also asserts *pll_unlock* to the GENERIC_FSK Link Layer hardware. GENERIC_FSK responds by asserting IRQ_CTRL[PLL_UNLOCK_IRQ] in GENERIC_FSK address space. GENERIC_FSK enters IN_WD state, which deasserts generic_fsk_rx_en to the TSM. This initiates the TSM TX warmdown. GENERIC_FSK holds in IN_WD to wait for TSM to return to idle. Once this occurs, GENERIC_FSK asserts IRQ_CTRL[SEQ_END_IRQ] and returns to its IDLE state. Three status bits are now set to indicate to software that the source of the abort was an RF_NOT_ALLOWED assertion:

- IRQSTS1[PLL_UNLOCK_IRQ]
- COEX_CTRL[RF_NOT_ALLOWED_ASSERTED]
- COEX_CTRL[RF_NOT_ALLOWED_RX_ABORT]

The complete set of control and status bits in the COEX_CTRL register is described in the following table:

Field	R/W	Description
RF_NOT_ALLOWED_EN	RW	The coexistence input RF_NOT_ALLOWED can be enabled to selectively abort TX or RX sequences, with individual enables for each supported protocol, subject also to the state of the RF_NOT_ALLOWED_NO_TX and RF_NOT_ALLOWED_NO_RX control bits, according to the table below. **NOTE:** RF_NOT_ALLOWED_EN[1] and RF_NOT_ALLOWED_EN[2] currently have no functionality **xxx1**: RF_NOT_ALLOWED assertions are enabled to abort BLE TX and RX sequences **xxx0**: RF_NOT_ALLOWED assertions are not enabled to abort BLE TX and RX sequences **1xxx**: RF_NOT_ALLOWED assertions are enabled to abort GENERIC_FSK TX and RX sequences **0xxx**: RF_NOT_ALLOWED assertions are not enabled to abort GENERIC_FSK TX and RX sequences
RF_NOT_ALLOWED_NO_TX	RW	TX Sequences will be aborted by RF_NOT_ALLOWED assertions, if the appropriate RF_NOT_ALLOWED_EN[x] bit is set.
RF_NOT_ALLOWED_NO_RX	RW	RX Sequences will be aborted by RF_NOT_ALLOWED assertions, if the appropriate RF_NOT_ALLOWED_EN[x] bit is set.
RF_NOT_ALLOWED_ASSERTED	R/W1TC	RF_NOT_ALLOWED has gone from not asserted, to asserted, since the last time this bit was cleared
RF_NOT_ALLOWED_TX_ABORT	R/W1TC	An TX sequence has been aborted since the last time this bit was cleared
RF_NOT_ALLOWED_TX_ABORT	R/W1TC	An TX sequence has been aborted since the last time this bit was cleared
RF_NOT_ALLOWED	R	Reflects the instantaneous state of the RF_NOT_ALLOWED pin,

Appendix A
Radio Register Summary

A.1 RSIM Memory Map and Register Definition

The RSIM memory map and description of registers are included in the following register section.

A.1.1 RSIM register descriptions

A.1.1.1 RSIM Memory map

RSIM base address: 4005_9000h

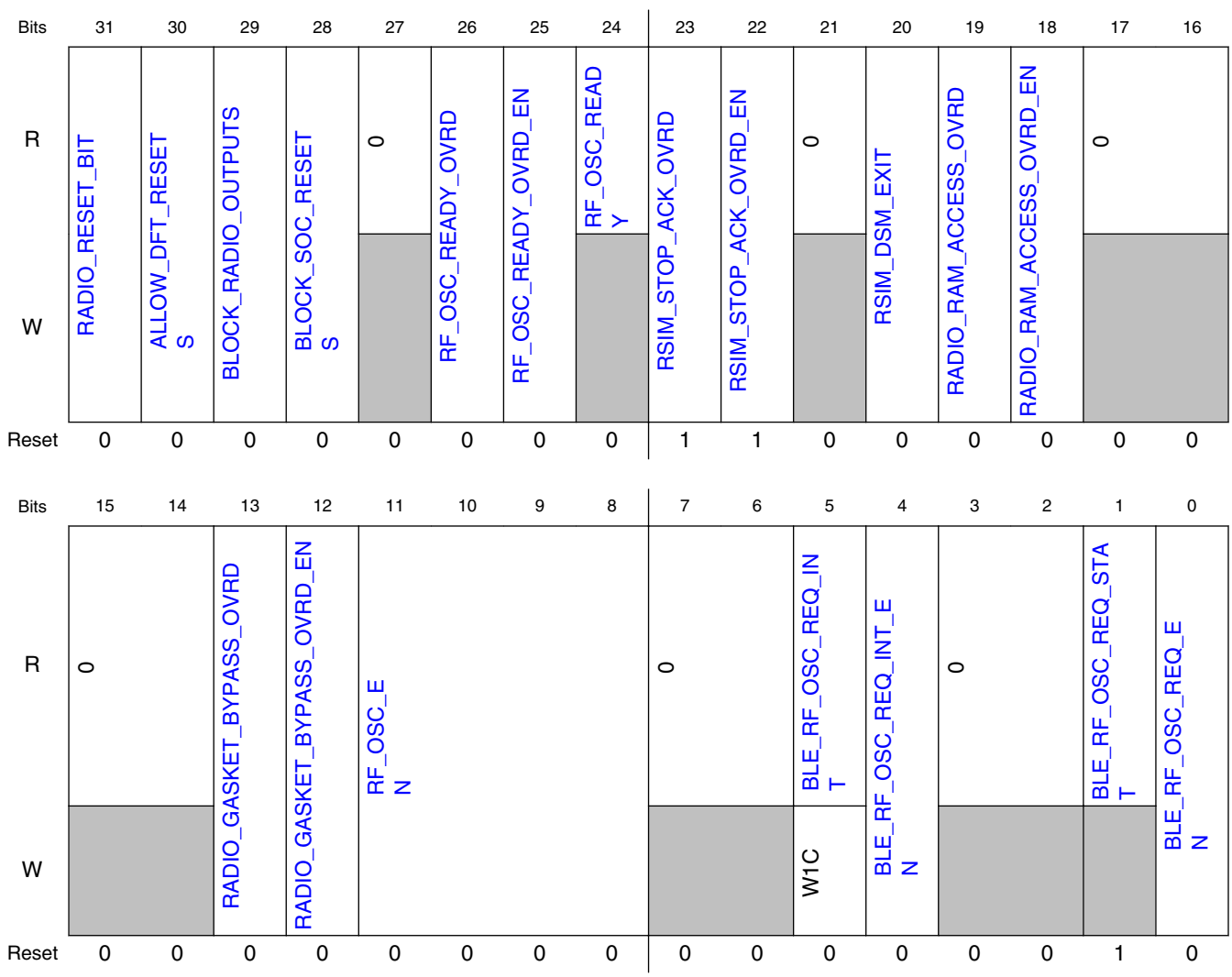
Offset	Register	Width (In bits)	Access	Reset value
0h	Radio System Control (CONTROL)	32	RW	00C0_0002h
4h	Deep Sleep Wakeup Sequence (DSM_WAKEUP)	32	RW	0000_0000h
8h	Radio MAC Address (MAC_MSB)	32	RO	0000_0000h
Ch	Radio MAC Address (MAC_LSB)	32	RO	0000_0000h
10h	Radio Miscellaneous (MISC)	32	RW	0800_0000h
18h	Radio Software Configuration (SW_CONFIG)	32	RW	0000_0020h
100h	Deep Sleep Timer (DSM_TIMER)	32	RO	0000_0000h
104h	Deep Sleep Timer Control (DSM_CONTROL)	32	RW	0000_0000h
108h	Deep Sleep Wakeup Time Offset (DSM_OSC_OFFSET)	32	RW	0000_0000h
11Ch	Generic FSK Link Layer Sleep Time (GEN_SLEEP)	32	RW	0000_0000h
120h	Generic FSK Link Layer Wake Time (GEN_WAKE)	32	RW	0000_0000h
124h	Radio Oscillator Control (RF_OSC_CTRL)	32	RW	A020_3806h
128h	Radio Analog Test Registers (ANA_TEST)	32	RW	0000_0000h
12Ch	Radio Analog Trim Registers (ANA_TRIM)	32	RW	784B_0000h

A.1.1.2 Radio System Control (CONTROL)

A.1.1.2.1 Offset

Register	Offset
CONTROL	0h

A.1.1.2.2 Diagram



A.1.1.2.3 Fields

Field	Function
31 RADIO_RESET_BIT	Software Reset for the Radio This bit resets on POR only. When the SoC 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 ALLOW_DFT_RESETS	Allow the DFT Reset Pin to Reset the Radio The Radio provides a port that can be connected to a test mode pin in order to provide a method of resetting the Radio independently from SoC resets. This bit enables that DFT functionality.
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_SOC_RESETS	Block SoC Resets of the Radio 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 SoC resets will be blocked and the Radio will not be affected by them when this bit is set.
27 —	Reserved
26 RF_OSC_READY_OVRD	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.
25 RF_OSC_READY_OVRD_EN	RF Ref Osc Ready Override Enable This bit enables the RF Ref Osc Ready Override bit.
24 RF_OSC_READY	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. The RF Ref Osc Ready signal can be overridden using the RF_OSC_READY_OVRD bit.
23 RSIM_STOP_ACK_OVRD	Stop Acknowledge Override This bit controls the Stop Acknowledge signal to the SoC Core Platform in Override mode.
22 RSIM_STOP_ACK_OVRD_EN	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 Deep Sleep Mode state of the Radio Link Layers and whether the Radio OSC is enabled by any means.
21 —	Reserved
20 RSIM_DSM_EXIT	BLE Force Deep Sleep Mode Exit This bit forces the BLE link layer to wakeup from Deep Sleep Mode.
19	Radio RAM Access Override

Table continues on the next page...

Field	Function
RADIO_RAM_ACCESS_OVRD	The Radio has two internal RAM blocks that are allowed to go into a low power state when they are not being used by a link layer. If this bit is set, then the RAMs are kept in an active power state to allow software to access them.
18 RADIO_RAM_ACCESS_OVRD_EN	Radio RAM Access Override Enable This bit enables the Radio RAM Access Override bit.
17-16 —	Reserved
15-14 —	Reserved
13 RADIO_GASKET_BYPASS_OVRD	Radio 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 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. 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. 0b - XCVR and Link Layer Register Clock is the RF Ref Osc Clock 1b - XCVR and Link Layer Register Clock is the SoC IPG Clock
12 RADIO_GASKET_BYPASS_OVRD_EN	Radio 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.
11-8 RF_OSC_EN	RF Ref Osc Enable Select The RF Reference Oscillator can be enabled by a Radio link layer, by an internal SoC clock mode, by an External Pin request, or by these bits. If these bits are all cleared, 0000, then the RF Ref Osc will be controlled by the SoC, by an external pin request, or by a 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. 0000b - RF Ref Osc will be controlled by the SoC, external pin, or a link layer 0001b - RF Ref Osc on in Run/Wait 0011b - RF Ref Osc on in Stop 0111b - RF Ref Osc on in VLPR/VLPW 1111b - RF Ref Osc on in VLPS
7-6 —	Reserved
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 Ref Osc (Sysclk) Request Interrupt Enable

Table continues on the next page...

Field	Function
BLE_RF_OSC_REQ_INT_EN	This bit enables an interrupt request when the enabled version of the BLE Ref Osc (Sysclk) Request is asserted high.
3-2 —	Reserved
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.

A.1.1.3 Deep Sleep Wakeup Sequence (DSM_WAKEUP)

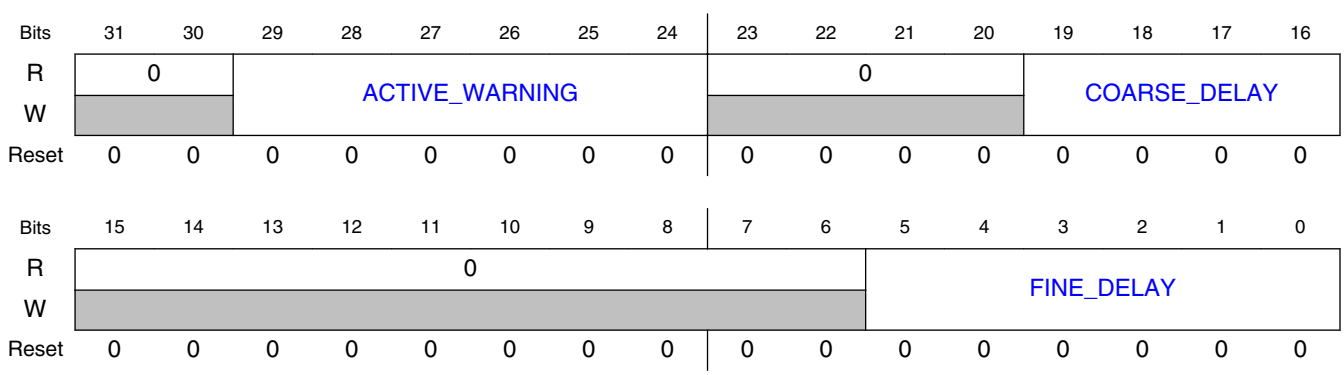
A.1.1.3.1 Offset

Register	Offset
DSM_WAKEUP	4h

A.1.1.3.2 Function

The RSIM Deep Sleep Wakeup Sequence register provides settings to adjust the RF Osc warm-up timing, and the delay of the RF Active signal used for WiFi coexistence.

A.1.1.3.3 Diagram



A.1.1.3.4 Fields

Field	Function
31-30 —	Reserved
29-24 ACTIVE_WARNING	Deep Sleep Wakeup RF Active Warning Time This is the delay after RF OSC has been enabled, the range is 0-63 and the resulting delay is 122.07 us x ACTIVE_WARNING + 30.5 us. This is typically when the RF Osc should be stable.
23-20 —	Reserved
19-16 COARSE_DELAY	Deep Sleep Wakeup Coarse Delay Time This is the first delay after the beginning of the Wake-Up sequence, the range is 0-15 and the resulting delay is 1.953 ms x COARSE_DELAY + 61 us, with a minimum delay of 61 us.
15-6 —	Reserved
5-0 FINE_DELAY	Deep Sleep Wakeup Fine Delay Time This is the second delay after the beginning of the Wake-Up sequence, the range is 0-63 and the resulting delay is 122.07 us x FINE_DELAY + 61 us, with a minimum delay of 61 us. This is when the RF OSC is enabled.

A.1.1.4 Radio MAC Address (MAC_MSB)

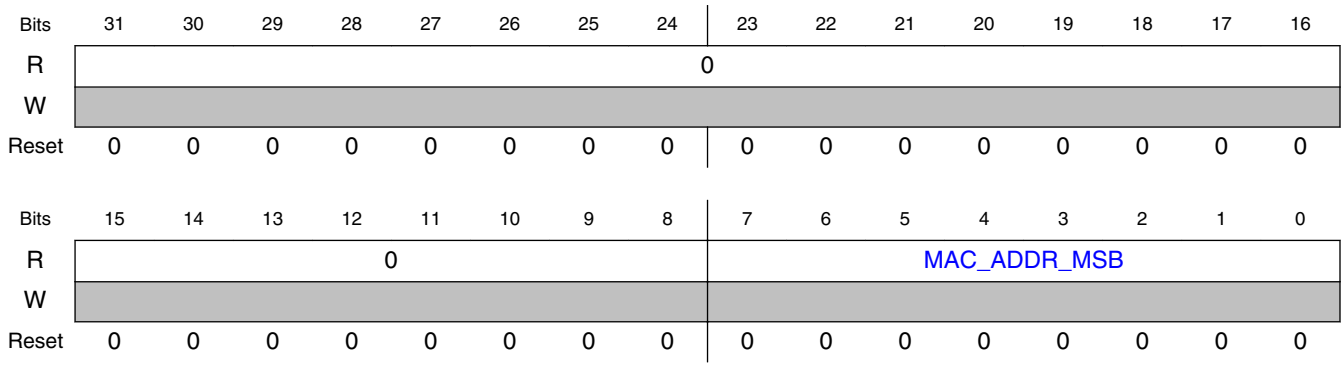
A.1.1.4.1 Offset

Register	Offset
MAC_MSB	8h

A.1.1.4.2 Function

The Radio MAC Address registers provide a unique ID that is stored in the Flash during factory test.

A.1.1.4.3 Diagram



A.1.1.4.4 Fields

Field	Function
31-8 —	Reserved
7-0 MAC_ADDR_MSB	Radio MAC Address MSB The Radio 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.

A.1.1.5 Radio MAC Address (MAC_LSB)

A.1.1.5.1 Offset

Register	Offset
MAC_LSB	Ch

A.1.1.5.2 Function

The Radio MAC Address registers provide a unique ID that is stored in the Flash during factory test

A.1.1.5.3 Diagram

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

A.1.1.5.4 Fields

Field	Function
31-0 MAC_ADDR_LSB	Radio MAC Address LSB The Radio 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.

A.1.1.6 Radio Miscellaneous (MISC)

A.1.1.6.1 Offset

Register	Offset
MISC	10h

A.1.1.6.2 Diagram

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

A.1.1.6.3 Fields

Field	Function
31-24 RADIO_VERSION	Radio Version ID number This register value can be read to identify the Version of the Radio. Note that the value will read as zero unless the Radio is in Run Mode. B = 2.4 GHz Radio 2.0 8 = 2.4 GHz Radio 2.1 4 = 2.4 GHz Radio 3.0 5 = 2.4 GHz Radio 3.1
23-20 —	Reserved
19-16 —	Reserved
15-5 —	Reserved
4-0 —	Reserved

A.1.1.7 Radio Software Configuration (SW_CONFIG)

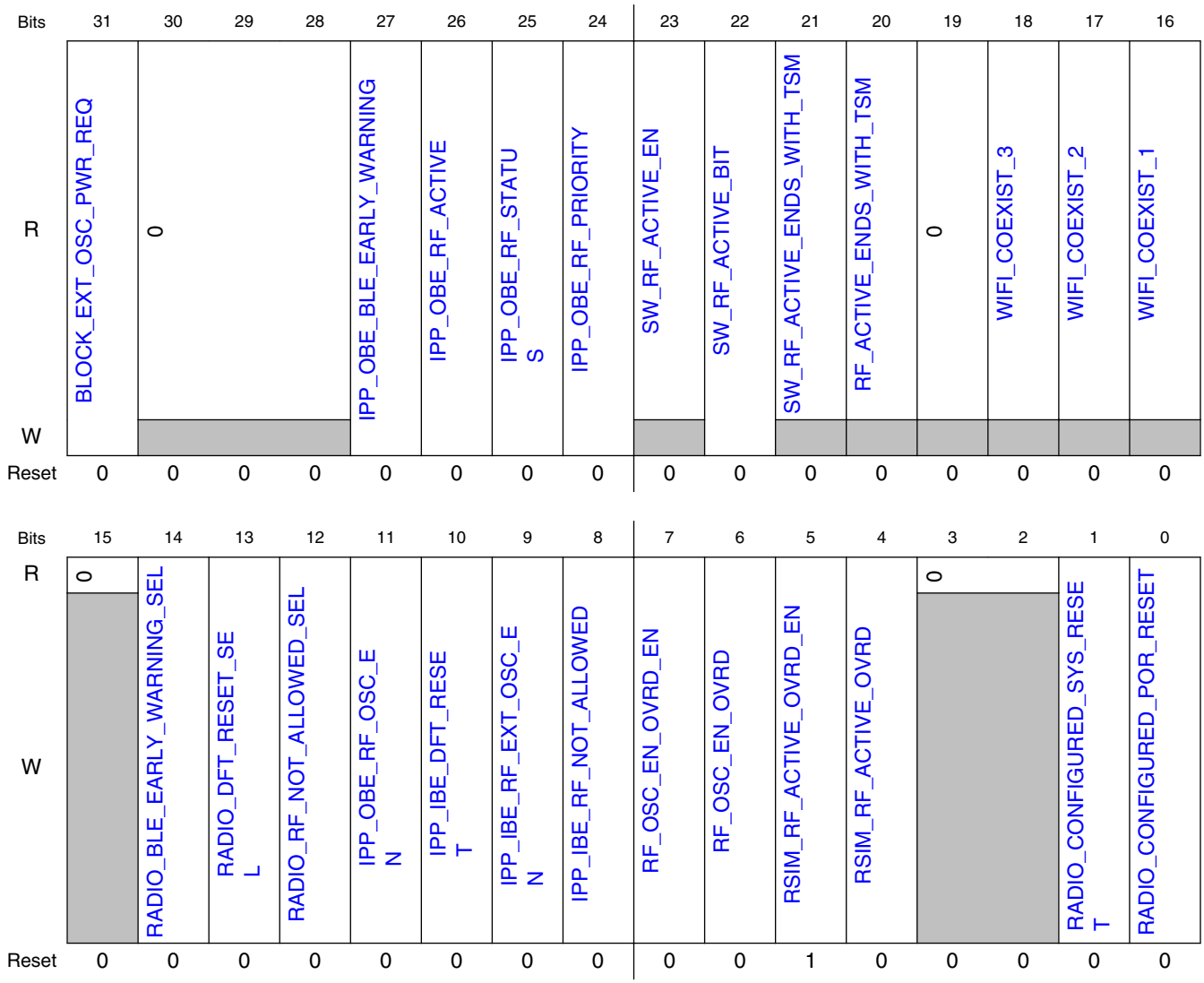
A.1.1.7.1 Offset

Register	Offset
SW_CONFIG	18h

A.1.1.7.2 Function

This register contains the software configuration and control bits.

A.1.1.7.3 Diagram



A.1.1.7.4 Fields

Field	Function
31 BLOCK_EXT_OSC_PWR_REQ	Block External Requests for RF OSC from starting a Radio Power Wakeup Sequence If this bit is set then any External request for the OSC will not start a Radio Power Wakeup Sequence, otherwise any such request does request full Run Regulation mode.
30-28 —	Reserved
27 IPP_OBE_BLE_EARLY_WARNING	IPP_OBE_BLE_EARLY_WARNING This bit enables the Output Driver (OBE) on the SoC port that provides the BLE Early Warning signal as a pin interface option.

Table continues on the next page...

Field	Function
26 IPP_OBE_RF_ACTIVE	IPP_OBE_RF_ACTIVE This bit enables the Output Driver (OBE) on the SoC port that provides the RF Active signal as a pin interface option.
25 IPP_OBE_RF_STATUS	IPP_OBE_RF_STATUS This bit enables the Output Driver (OBE) on the SoC port that provides the RF Status signal as a pin interface option.
24 IPP_OBE_RF_PRIORITY	IPP_OBE_RF_PRIORITY This bit enables the Output Driver (OBE) on the SoC port that provides the RF Priority signal as a pin interface option.
23 SW_RF_ACTIVE_EN	Software RF_ACTIVE Control Enable If this bit is set the SW_RF_ACTIVE_BIT acts as an OR source for the RF_ACTIVE pin.
22 SW_RF_ACTIVE_BIT	Software RF_ACTIVE Control Bit If the SW_RF_ACTIVE_EN bit is set this bit acts as an OR source for the RF_ACTIVE pin.
21 SW_RF_ACTIVE_ENDS_WITH_TSM	Software RF_ACTIVE clearing mechanism If this bit is set along with the SW_RF_ACTIVE_EN bit, then the Software RF_ACTIVE OR source will clear when the TSM sequence ends.
20 RF_ACTIVE_ENDS_WITH_TSM	RF_ACTIVE clearing mechanism If this bit is set then the RSIM RF_ACTIVE OR source will clear when the TSM sequence ends.
19 —	Reserved
18 WIFI_COEXIST_3	RF_EARLY_WARNING Source If this bit is set the RF_EARLY_WARNING pin will be controlled by TSM Spare 3, otherwise it will be controlled by the BLE Link Layer.
17 WIFI_COEXIST_2	RF_STATUS Source If this bit is set the RF_STATUS pin will be controlled by TSM Spare 3, otherwise it will be controlled by TSM Spare 2.
16 WIFI_COEXIST_1	RF_ACTIVE Source If this bit is set the RF_ACTIVE pin will be controlled by TSM Spare 1, otherwise it will be controlled by the RSIM RF Active Sequence along with any override by the SW_RF_ACTIVE_BIT.
15 —	Reserved
14 RADIO_BLE_EARLY_WARNING_SEL	Radio BLE_EARLY_WARNING Select The Radio has two pins which may be used as an early warning of a BLE activity. This bit selects which of the two pins has that early warning.
13 RADIO_DFT_RESET_SEL	Radio DFT_RESET Select The Radio has two pins which may be used as a DFT_RESET. This bit selects the two pins.
12	Radio RF_NOT_ALLOWED Select

Table continues on the next page...

Field	Function
RADIO_RF_NOT_ALLOWED_SEL	The Radio has two pins which may be controlled by external pins in the SoC in such a way as to allow an external device, such as another radio, to coexist with this Radio. This bit selects which of the two pins has control of that external request.
11 IPP_OBE_RF_OSC_EN	IPP_OBE_RF_OSC_EN This bit enables the Output Driver (OBE) on the SoC port that provides the RF Osc Enable signal as a pin interface option.
10 IPP_IBE_DFT_RESET	IPP_IBE_DFT_RESET This bit enables the Input Driver (IBE) on the SoC port that provides the DFT_RESET as a pin interface option.
9 IPP_IBE_RF_EXT_OSC_EN	IPP_IBE_RF_EXT_OSC_EN This bit enables the Input Driver (IBE) on the SoC port that provides the RF_EXT_OSC_EN as a pin interface option.
8 IPP_IBE_RF_NOT_ALLOWED	IPP_IBE_RF_NOT_ALLOWED This bit enables the Input Driver (IBE) on the SoC port that provides the RF_NOT_ALLOWED as a pin interface option.
7 RF_OSC_EN_OVRD_EN	Radio Osc Enable Override Enable This bit enables the Radio Osc Enable Override bit.
6 RF_OSC_EN_OVRD	Radio Osc Enable Override This bit directly controls the RF_OSC_EN SoC pin signal when the Radio Osc Enable Override is enabled.
5 RSIM_RF_ACTIVE_OVRD_EN	RF Active Internal Override Enable This bit enables the RF Active Internal Override bit.
4 RSIM_RF_ACTIVE_OVRD	RF Active Internal Override If the RSIM_RF_ACTIVE_OVRD_EN bit is set, then this bit controls the Internal version of RF Active instead of the Link Layers. The Internal version of RF Active whether overridden or not has a lower priority and can always be overridden by software using the SW_RF_ACTIVE_BIT or the TSM Spare 1 bit if it is selected using the WIFI_COEXIST_1 select bit.
3-2 —	Reserved
1 RADIO_CONFIGURED_SYS_RESET	Radio Configuration Bit, cleared by Radio System Reset This is a bit software can write when it wants to be able to check if a Radio System Reset occurs.
0 RADIO_CONFIGURED_POR_RESET	Radio Configuration Bit, cleared by Radio Power On Reset This is a bit software can write when it wants to be able to check if a Radio Power On Reset occurs.

A.1.1.8 Deep Sleep Timer (DSM_TIMER)

A.1.1.8.1 Offset

Register	Offset
DSM_TIMER	100h

A.1.1.8.2 Diagram

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

A.1.1.8.3 Fields

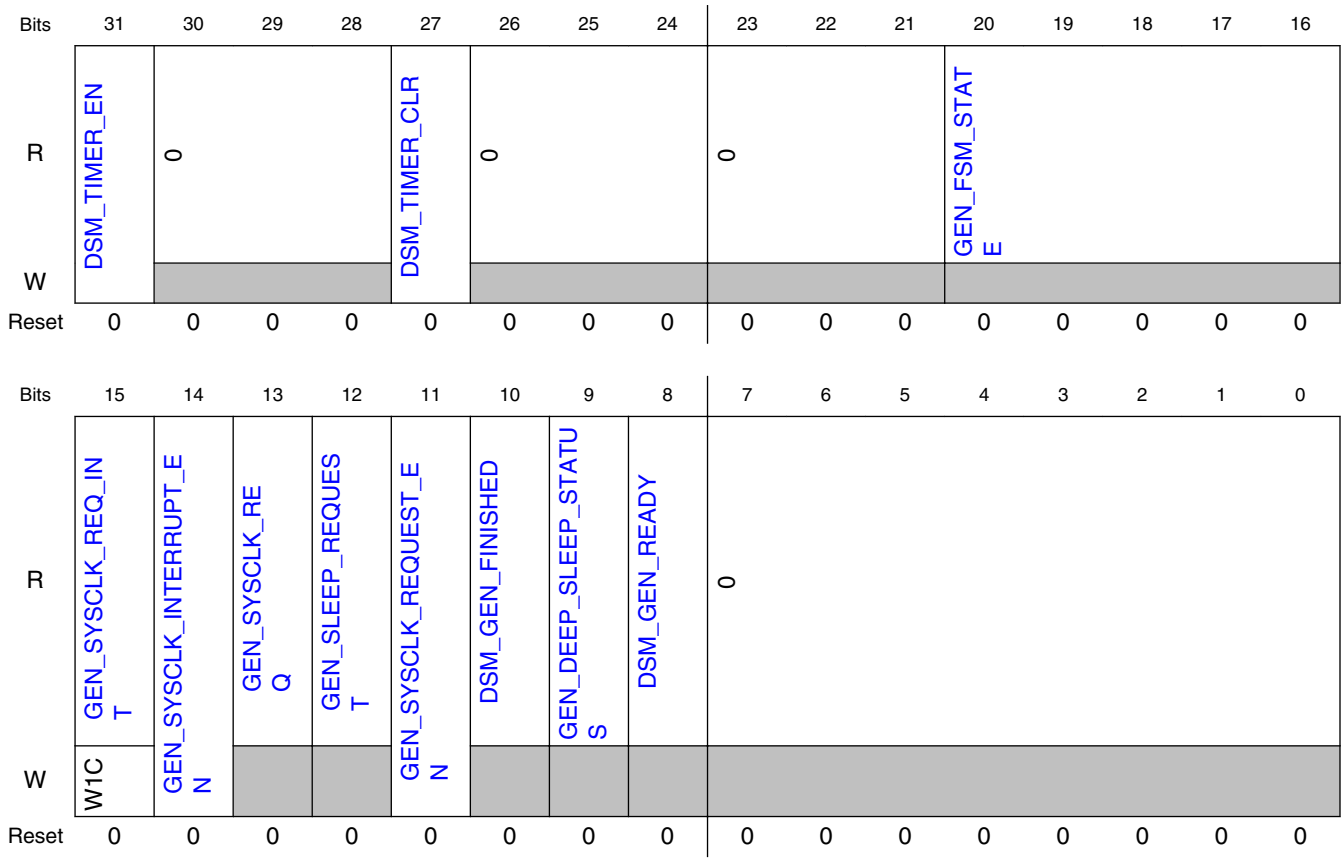
Field	Function
31-24 —	Reserved
23-0 DSM_TIMER	Deep Sleep Mode Timer This read only register shows the value of the Deep Sleep Timer. The timer counts clock cycles of the 32 kHz SoC oscillator whenever the DSM_TIMER_EN bit is set. The timer is reset to zero whenever the DSM_TIMER_CLR bit is set.

A.1.1.9 Deep Sleep Timer Control (DSM_CONTROL)

A.1.1.9.1 Offset

Register	Offset
DSM_CONTROL	104h

A.1.1.9.2 Diagram



A.1.1.9.3 Fields

Field	Function
31 DSM_TIMER_EN	Deep Sleep Mode Timer Enable Whenever this bit is set the Deep Sleep Mode Timer counts clock cycles of the 32 kHz SoC oscillator.
30-28 —	Reserved
27 DSM_TIMER_CLR	Deep Sleep Mode Timer Clear Whenever this bit is set the Deep Sleep Mode Timer is reset to zero.
26-24 —	Reserved
23-21 —	Reserved
20-16 GEN_FSM_STATE	GEN Deep Sleep State Machine State This is the current State that the GEN Deep Sleep Finite State Machine is in.

Table continues on the next page...

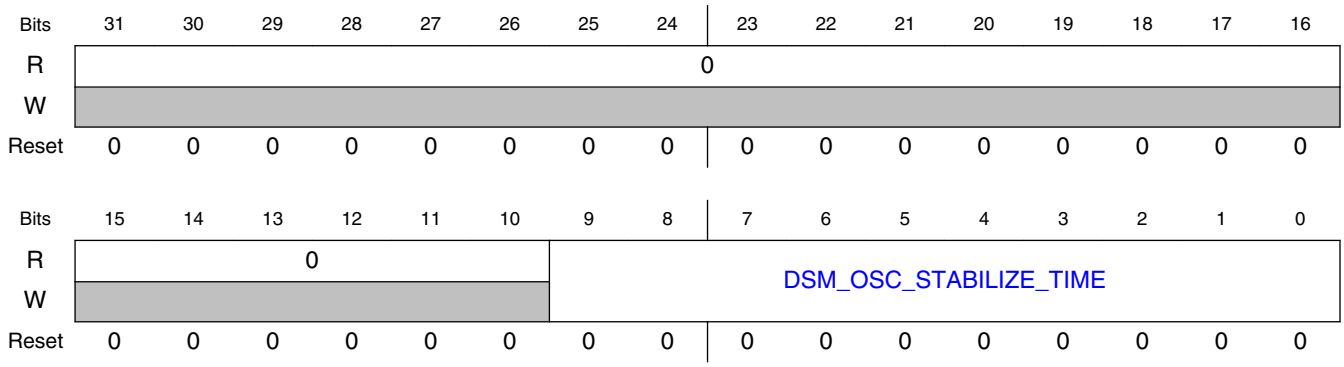
Field	Function
15 GEN_SYSCLK_REQ_INT	Interrupt Flag from an Generic FSK Link Layer RF OSC Request If the GEN_SYSCLK_INTERRUPT_EN bit is set, then this flag indicates that an Generic FSK RF OSC Request has occurred. The flag will persist until this bit is written with a 1 to clear it.
14 GEN_SYSCLK_INTERRUPT_EN	Generic FSK Link Layer RF OSC Request Interrupt Enable If this bit is set, then a change from a 0 to a 1 on GEN_SYSCLK_REQ will generate an interrupt which can be used by the SoC.
13 GEN_SYSCLK_REQ	Generic FSK Link Layer RF OSC Request Status This bit shows the status of the Generic FSK Link Layer Request for the RF OSC to be turned on. If the GEN_SYSCLK_REQUEST_EN bit is set, then this signal is used by the RSIM to turn on the RF OSC if the RF OSC is not already turned on by another request source.
12 GEN_SLEEP_REQUEST	Generic FSK Link Layer Deep Sleep Requested From the Generic FSK Sleep Enable register. This enables a match of GEN_SLEEP[23:0] to SLEEP_TMR[23:0]; when the match occurs Deep Sleep Mode is entered.
11 GEN_SYSCLK_REQUEST_EN	Enable Generic FSK Link Layer to Request RF OSC This bit allows the Generic FSK Link Layer to request turning on the RF OSC.
10 DSM_GEN_FINISHED	Generic FSK Deep Sleep Time Finished This is the SLEEP_FINISHED state in the Deep Sleep Module State Machine, the state machine holds here until the Link Layer de-asserts the sleep_request.
9 GEN_DEEP_SLEEP_STATUS	Generic FSK Link Layer Deep Sleep Mode Status This is the signal from the Deep Sleep Module State Machine telling the Link Layer to enter and exit Deep Sleep Mode. If this bit is set then the Link Layer is in Deep Sleep Mode.
8 DSM_GEN_READY	Generic FSK Ready for Deep Sleep Mode This is the SLEEP_READY state in the Deep Sleep Module State Machine, the state machine holds here until the sleep time is reached on the 32 kHz clock counter (Deep Sleep Timer).
7-0 —	Reserved

A.1.1.10 Deep Sleep Wakeup Time Offset (DSM_OSC_OFFSET)

A.1.1.10.1 Offset

Register	Offset
DSM_OSC_OFFSET	108h

A.1.1.10.2 Diagram



A.1.1.10.3 Fields

Field	Function
31-10 —	Reserved
9-0 DSM_OSC_STABILIZE_TIME	Deep Sleep Wakeup RF OSC Stabilize Time This register should be programmed by software with the time needed for the RF OSC to stabilize to its specified accuracy after it is enabled. This time is represented by the number of clock cycles of the 32 kHz SoC oscillator that should be counted before the link layer is allowed to exit Deep Sleep Mode on a wakeup event. This time offset will be used by the Deep Sleep State Machine to first turn on the RF OSC, then wait for it stabilize, and finally wakeup the link layer.

A.1.1.11 Generic FSK Link Layer Sleep Time (GEN_SLEEP)

A.1.1.11.1 Offset

Register	Offset
GEN_SLEEP	11Ch

A.1.1.11.2 Diagram

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

A.1.1.11.3 Fields

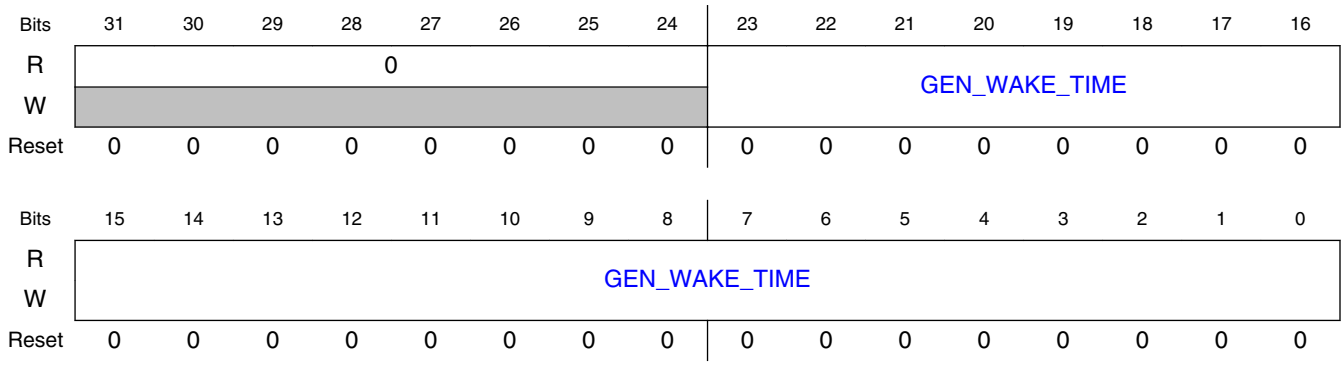
Field	Function
31-24 —	Reserved
23-0 GEN_SLEEP_TIME	Generic FSK Link Layer Sleep Time Software determines the future time at which it would like to enter Deep Sleep Mode by reading the DSM_TIMER, computing the number of 32 kHz clock cycles remaining until the desired DSM start-time, and writing that value into this register. The value programmed into this register should be no fewer than 4 clocks greater (in the future) than the current time as read from DSM_TIMER.

A.1.1.12 Generic FSK Link Layer Wake Time (GEN_WAKE)

A.1.1.12.1 Offset

Register	Offset
GEN_WAKE	120h

A.1.1.12.2 Diagram



A.1.1.12.3 Fields

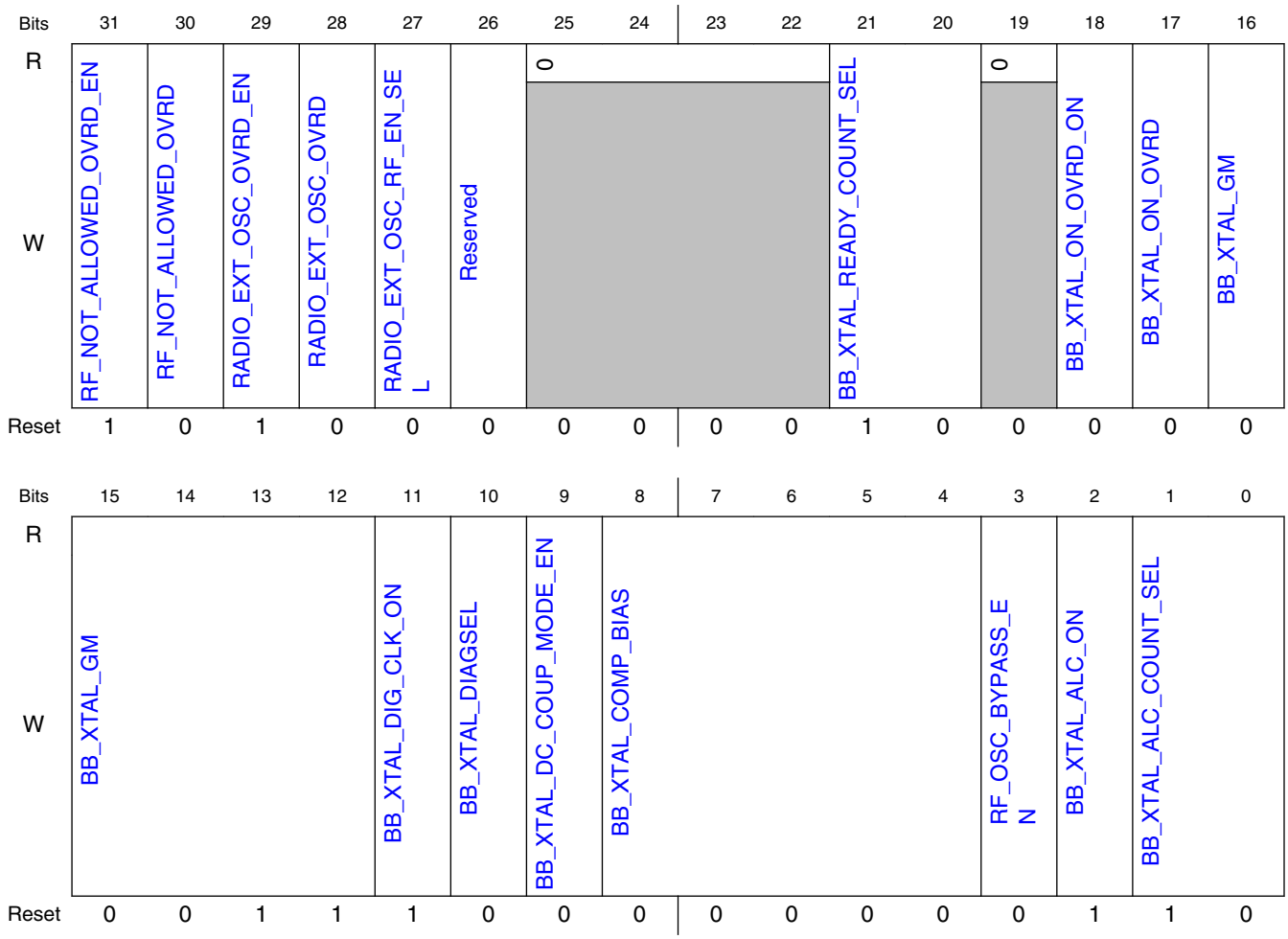
Field	Function
31-24 —	Reserved
23-0 GEN_WAKE_TIME	Generic FSK Link Layer Wake Time Software determines the future time at which it would like to exit Deep Sleep Mode by reading the DSM_TIMER, computing the number of 32 kHz clock cycles remaining until the desired DSM exit time, and writing that value into this register.

A.1.1.13 Radio Oscillator Control (RF_OSC_CTRL)

A.1.1.13.1 Offset

Register	Offset
RF_OSC_CTRL	124h

A.1.1.13.2 Diagram



A.1.1.13.3 Fields

Field	Function
31 RF_NOT_ALLOWED_OVRD_EN	RF Not Allowed Override Enable This bit enables the RF Not Allowed Override bit.
30 RF_NOT_ALLOWED_OVRD	RF Not Allowed Override If the RF_NOT_ALLOWED_OVRD_EN bit is set, then this bit controls the RF Not Allowed functionality instead of the SoC pin.
29 RADIO_EXT_OSC_OVRD_EN	Radio External Request for RF OSC Override Enable This bit enables the Radio External RF OSC Override bit.
28	Radio External Request for RF OSC Override

Table continues on the next page...

Field	Function
RADIO_EXT_OSC_OVRD	If the RADIO_EXT_OSC_OVRD_EN bit is set, then this bit controls that External RF OSC Request functionality.
27 RADIO_EXT_OSC_RF_EN_SEL	Radio External Request for RF OSC Select The Radio has two pins which may be controlled by external pins in the SoC in such a way as to allow an external device, such as another radio, to enable this Radio's RF Oscillator. This bit selects which of the two pins has control of that external request.
26 —	Reserved
25-22 —	Reserved
21-20 BB_XTAL_READY_COUNT_SEL	rmap_bb_xtal_ready_count_sel_hv[1:0] Program counter for xtal ready signal Sets up count value for XO startup time. 00b - 1024 counts (32 us @ 32 MHz) 01b - 2048 (64 us @ 32 MHz) 10b - 4096 (128 us @ 32 MHz) 11b - 8192 (256 us @ 32 MHz)
19 —	Reserved
18 BB_XTAL_ON_OVRD_ON	rmap_bb_xtal_on_ovrd_on_hv Enable override XO enable bit Enable selector: 0b - rfctrl_bb_xtal_on_hv is asserted 1b - rfctrl_bb_xtal_on_ovrd_hv is asserted
17 BB_XTAL_ON_OVRD	rmap_bb_xtal_on_ovrd_hv Override XO enable
16-12 BB_XTAL_GM	rmap_bb_xtal_gm_hv[4:0] Amplifier current bumps, bit [4] not used. Current values assume ALC is off 0 - Min setting: 8 current sources on (TT27:60uA) ... 15 - Max setting: 128 current sources on (TT27:960uA)
11 BB_XTAL_DIG_CLK_ON	rmap_bb_xtal_dig_clk_on_hv Enable digital clk output
10 BB_XTAL_DIAGSEL	rmap_bb_xtal_diagsel_hv Enable diagnostics for XO block
9 BB_XTAL_DC_COUP_MODE_EN	rmap_bb_xtal_dc_coup_mode_en_hv This bit enables the external dc coupled mode. This bit powers down the XO amplifier to enable DC coupled input.
8-4	rmap_bb_xtal_comp_bias_hv[4:0]

Table continues on the next page...

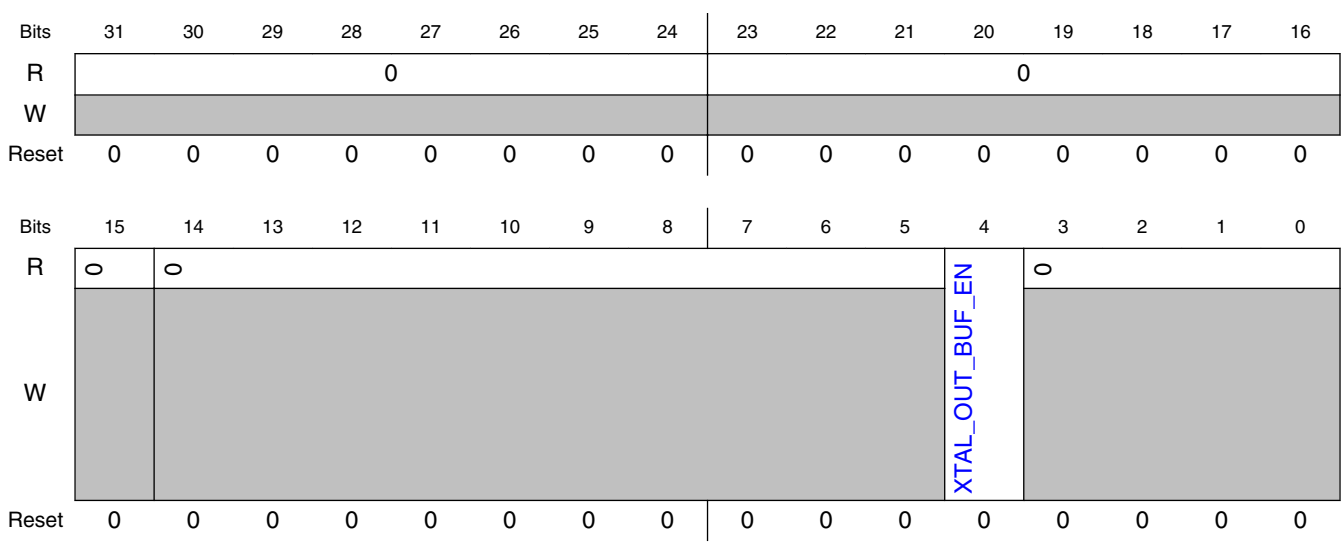
Field	Function
BB_XTAL_COMP_BIAS	Not used.
3 RF_OSC_BYPASS_EN	RF Ref Osc Bypass Enable This bit engages the RF Ref Osc analog bypass circuit if the RF Ref Osc is enabled. When the RF Ref Osc is in bypass mode it passes the RF EXTAL clock as the RF Ref Osc clock. Note that the RF Ref Osc Ready signal functions normally in RF OSC Bypass mode, unless overridden with the RF_OSC_READY_OVRD_EN bit.
2 BB_XTAL_ALC_ON	rmap_bb_xtal_alc_on_hv Enable ALC
1-0 BB_XTAL_ALC_COUNT_SEL	rmap_bb_xtal_alc_count_sel_hv[1:0] Program counter for alc ready signal Sets up count value for fastcharge to turn off: 00b - 2048 (64 us @ 32 MHz) 01b - 4096 (128 us @ 32 MHz) 10b - 8192 (256 us @ 32 MHz) 11b - 16384 (512 us @ 32 MHz)

A.1.1.14 Radio Analog Test Registers (ANA_TEST)

A.1.1.14.1 Offset

Register	Offset
ANA_TEST	128h

A.1.1.14.2 Diagram



A.1.1.14.3 Fields

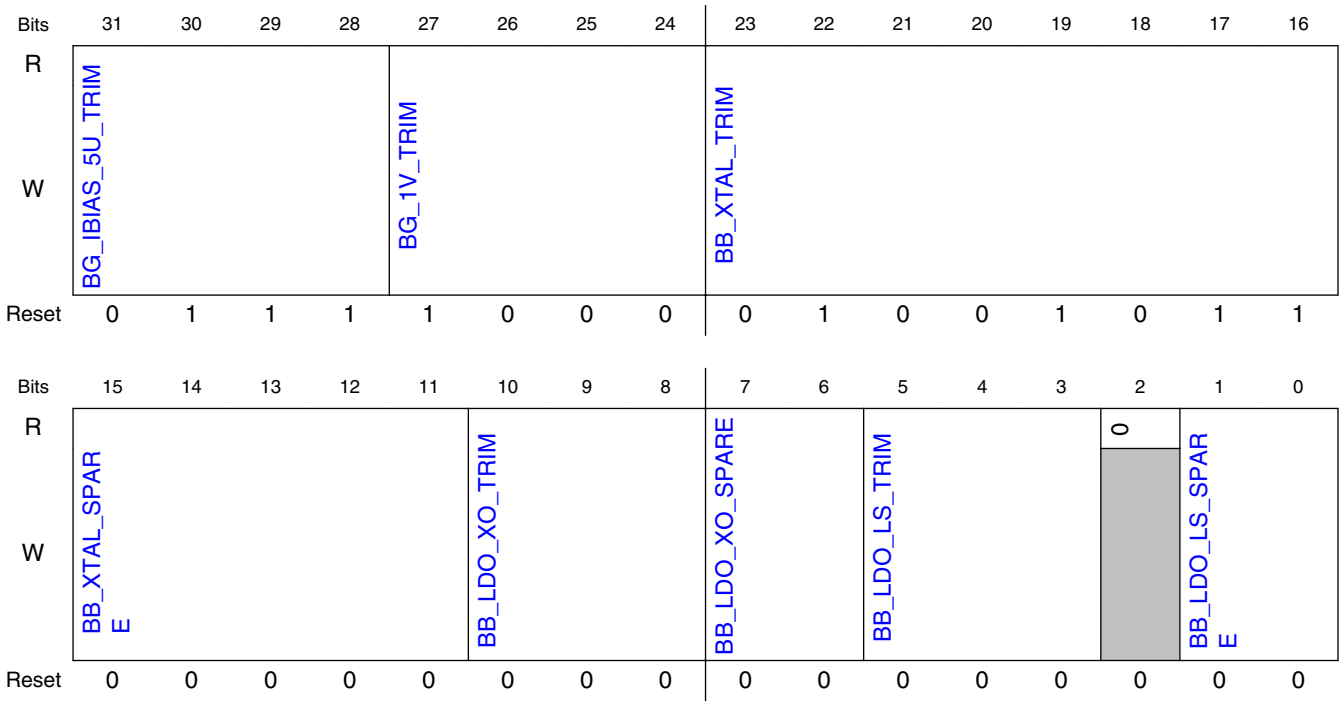
Field	Function
31-24 —	Reserved
23-15 —	Reserved
14-5 —	Reserved.
4 XTAL_OUT_BUF_EN	XTAL Output Buffer Enable The RF Analog Oscillator XTAL_OUT buffer for the XTAL_OUT SoC Pin is enabled when this bit is set. Note that this bit only enables the buffer to the external pin, it does not enable XTAL operation. That must be done through either an internal SOC request or external request via RF_RFOSC_EN pin.
3-0 —	Reserved.

A.1.1.15 Radio Analog Trim Registers (ANA_TRIM)

A.1.1.15.1 Offset

Register	Offset
ANA_TRIM	12Ch

A.1.1.15.2 Diagram



A.1.1.15.3 Fields

Field	Function
31-28 BG_IBIAS_5U_TRIM	rmap_bg_ibias_5u_trim_hv[3:0] 5uA current trim bits. Default setting is 0111 Trim bits for VBG output current 0000b - 3.55 uA 0001b - 3.73 uA 0010b - 4.04 uA 0011b - 4.22 uA 0100b - 4.39 uA 0101b - 4.57 uA 0110b - 4.89 uA 0111b - 5.06 (Default) 1000b - 5.23 uA 1001b - 5.41 uA 1010b - 5.72 uA 1011b - 5.9 uA 1100b - 6.07 uA 1101b - 6.25 uA 1110b - 6.56 uA 1111b - 6.74 uA
27-24 BG_1V_TRIM	rmap_bg_1v_trim_hv[3:0] Trim bits for VBG output voltage Trim bits for VBG output voltage

Table continues on the next page...

Field	Function
	0000b - 954.14 mV 0001b - 959.26 mV 0010b - 964.38 mV 0011b - 969.5 mV 0100b - 974.6 mV 0101b - 979.7 mV 0110b - 984.8 mV 0111b - 989.9 mV 1000b - 995 mV (Default) 1001b - 1 V 1010b - 1.005 V 1011b - 1.01 V 1100b - 1.015 V 1101b - 1.02 V 1110b - 1.025 V 1111b - 1.031 V
23-16 BB_XTAL_TRIM	rmap_bb_xtal_trim_hv[7:0] Bump XO load capacitor Load capacitor bumps. bit [7] not used. 0 - Min C1: 5.7pF. Min C2: 7.1pF ... 127 - Max C1: 22.6pF. Max C2: 28.2pF
15-11 BB_XTAL_SPARE	rmap_bb_xtal_spare_hv[4:0] Bit 4: Override RF Analog XTAL Ready signal ; 0 - No override, BB_XTAL_READY_COUNT_SEL is used ; 1- Force RF Analog XTAL Ready = 1. Note that this RF Analog signal can be also be overridden using the RSIM RF_OSC_READY_OVRD bits. Bit 3: XTAL_OUT output Polarity invert bit; 0 - No Polarity inversion; 1 - Polarity inverted Bit 2: This bit selects the clock to be output on the XTAL_OUT SoC pin. The default is to select the XTAL as the clock source. If this bit is written to a 1 then the AuxPLL output will be selected instead. Bits 1:0 are unused.
10-8 BB_LDO_XO_TRIM	rmap_bb_ldo_xo_trim_hv[2:0] Trim settings for LDO. Trim settings for LDO. 000b - 1.20 V (Default) 001b - 1.25 V 010b - 1.28 V 011b - 1.33 V 100b - 1.40 V 101b - 1.44 V 110b - 1.50 V 111b - 1.66 V
7-6 BB_LDO_XO_SPARE	rmap_bb_ldo_xo_spare_hv[1:0] Spare bits for LDO, not used.
5-3 BB_LDO_LS_TRIM	rmap_bb_ldo_ls_trim_hv[2:0] Trim settings for LDO. Trim settings for LDO.

Table continues on the next page...

Field	Function
	000b - 1.20 V (Default) 001b - 1.25 V 010b - 1.28 V 011b - 1.33 V 100b - 1.40 V 101b - 1.44 V 110b - 1.50 V 111b - 1.66 V
2 —	Reserved
1-0 BB_LDO_LS_SPARE	rmap_bb_ldo_ls_spare_hv[1:0] Spare bits. Not used so far.

A.2 Transceiver Memory Map and Register Definition

The transceiver memory map and description of registers are included in the following register section.

A.2.1 XCVR_TX_DIG register descriptions

A.2.1.1 XCVR_TX_DIG_ADDR Memory map

XCVR_TX_DIG base address: 4005_C200h

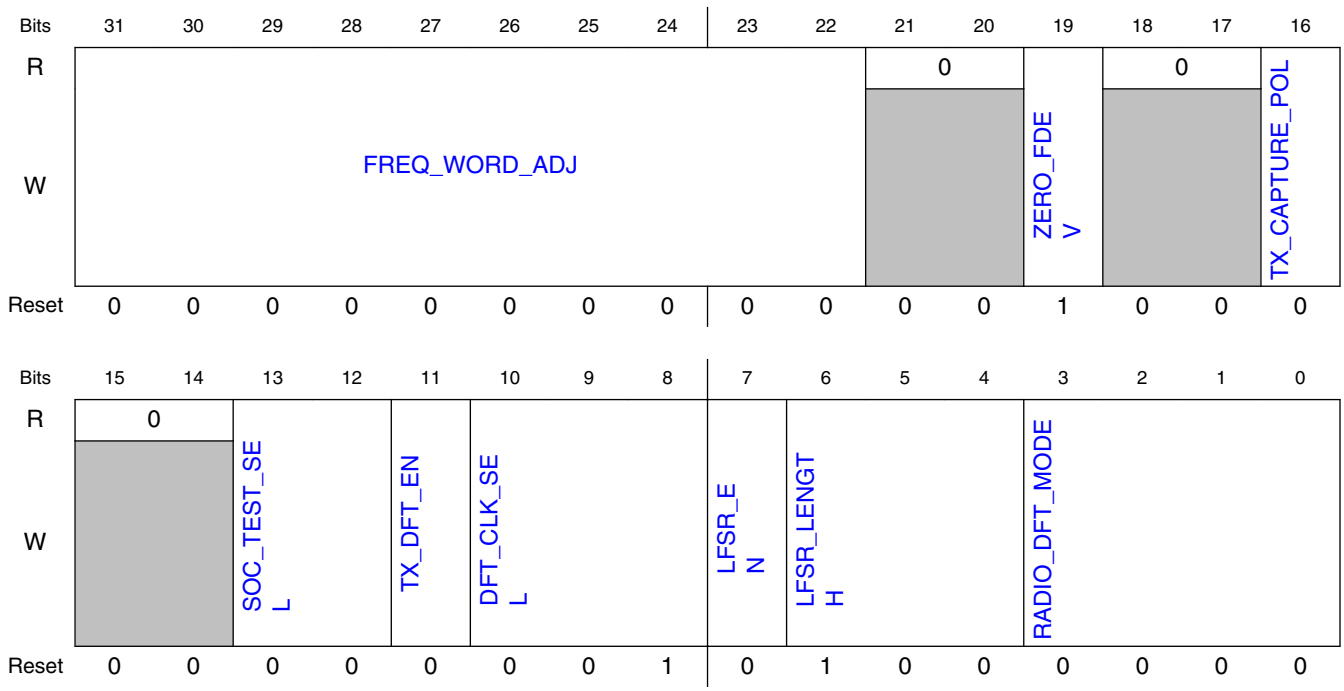
Offset	Register	Width (In bits)	Access	Reset value
0h	TX Digital Control (CTRL)	32	RW	0008_0140h
4h	TX Data Padding (DATA_PADDING)	32	RW	7FFF_55AAh
8h	TX GFSK Modulator Control (GFSK_CTRL)	32	RW	0001_4000h
Ch	TX GFSK Filter Coefficients 2 (GFSK_COEFF2)	32	RW	C063_0401h
10h	TX GFSK Filter Coefficients 1 (GFSK_COEFF1)	32	RW	BB29_960Dh
14h	TX FSK Modulation Levels (FSK_SCALE)	32	RW	0800_1800h
18h	TX DFT Modulation Pattern (DFT_PATTERN)	32	RW	0000_0000h

A.2.1.2 TX Digital Control (CTRL)

A.2.1.2.1 Offset

Register	Offset
CTRL	0h

A.2.1.2.2 Diagram



A.2.1.2.3 Fields

Field	Function
31-22 FREQ_WORD_ADJ	Frequency Word Adjustment This register is a signed 9 bit number that is added to the TX Digital output before it is presented to the PLL as the baseband frequency word. This allows the baseband frequency word to be adjusted, or skewed, by a range of -512 to +511. This frequency adjustment is applied to the final modulation word from any source (GFSK, FSK, DFT) and there is no protection from a math overflow. So the baseband frequency word range of -4096 to 4095 must be considered when adding this frequency word adjustment.
21-20 —	Reserved
19	On-Air at Zero FDev

Table continues on the next page...

Field	Function
ZERO_FDEV	If this bit is set, then in Gaussian Modulation modes the modulation will start at the Carrier Frequency instead of at the negative modulation value.
18-17 —	Reserved
16 TX_CAPTURE_POL	Polarity of the Input Data for the Transmitter If this bit is set, the TX data presented to the Transmitter will be inverted before it is processed.
15-14 —	Reserved
13-12 SOC_TEST_SEL	Radio Clock Selector for SoC RF Clock Tests This register selects the Radio clock source for the SoC Clock Tests Frequency Measurement. 00b - No Clock Selected 01b - PLL Sigma Delta Clock, divided by 2 10b - Auxiliary PLL Clock, divided by 2 11b - RF Ref Osc clock, divided by 2
11 TX_DFT_EN	DFT Modulation Enable If the Radio is in a DFT Pattern Register mode, then this bit is used to turn on and off the modulation that is shifted out from the pattern register.
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 in DFT Pattern Register modes, and the same frequency is also used to clock the LFSR and generate the pseudo-random modulation in DFT LFSR modes. 000b - 62.5 kHz 001b - 125 kHz 010b - 250 kHz 011b - 500 kHz 100b - 1 MHz 101b - 2 MHz 110b - 4 MHz 111b - RF OSC Clock
7 LFSR_EN	LFSR Enable If the Radio is in a DFT LFSR mode, then this bit is used to turn on and off the LFSR that is used to generate the modulation. Note that the LFSR is clocked at the DFT Clock frequency.
6-4 LFSR_LENGTH	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] 000b - LFSR 9, tap mask 100010000 001b - LFSR 10, tap mask 1001000000 010b - LFSR 11, tap mask 11101000000 011b - LFSR 13, tap mask 1101100000000 100b - LFSR 15, tap mask 111010000000000 101b - LFSR 17, tap mask 11110000000000000 110b - Reserved 111b - Reserved
3-0	Radio DFT Modes

Field	Function
RADIO_DFT_MODE	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. For LFSR modes the LFSR_EN needs to be set to turn on the LFSR. 0000b - Normal Radio Operation, DFT not engaged. 0001b - Carrier Frequency Only 0010b - Pattern Register GFSK 0011b - LFSR GFSK 0100b - Pattern Register FSK 0101b - LFSR FSK 0110b - Pattern Register O-QPSK 0111b - LFSR O-QPSK 1000b - LFSR 802.15.4 Symbols 1001b - PLL Modulation from RAM 1010b - PLL Coarse Tune BIST 1011b - PLL Frequency Synthesizer BIST 1100b - High Port DAC BIST 1101b - VCO Frequency Meter 1110b - Reserved 1111b - Reserved

A.2.1.3 TX Data Padding (DATA_PADDING)

A.2.1.3.1 Offset

Register	Offset
DATA_PADDING	4h

A.2.1.3.2 Diagram

Bits	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

Bits	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

A.2.1.3.3 Fields

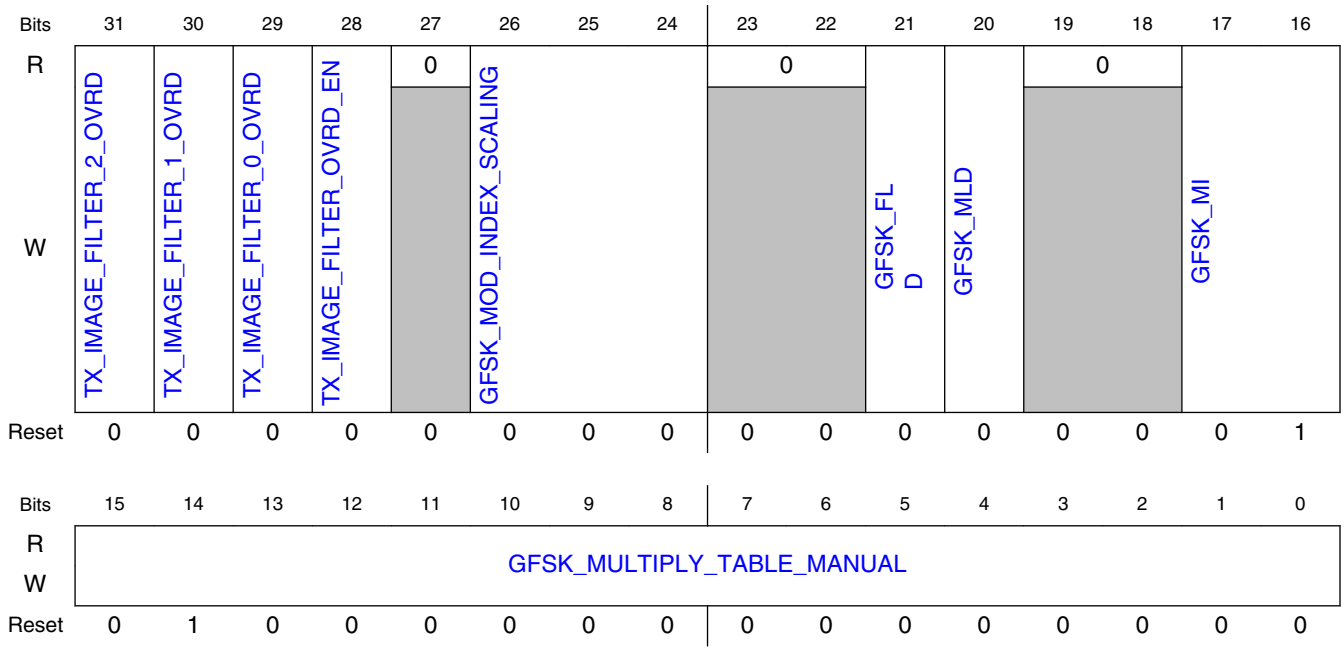
Field	Function
31 LRM	LFSR Reset Mask When this bit is set the DFT LFSR will not be reset when LFSR_EN is cleared and will instead continue to repeat its sequence as defined by the DFT_LFSR_LEN bits when LFSR_EN is next set. When this bit is cleared the DFT LFSR will reset every time LFSR_EN is cleared.
30-16 DFT_LFSR_OUT	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 the first bit of the Preamble is 1; the LSB is the first bit shifted out as padding.
7-0 DATA_PADDING_PAT_0	Data Padding Pattern 0 These bits are used for Data Padding when the first bit of the Preamble is 0; the LSB is the first bit shifted out as padding.

A.2.1.4 TX GFSK Modulator Control (GFSK_CTRL)

A.2.1.4.1 Offset

Register	Offset
GFSK_CTRL	8h

A.2.1.4.2 Diagram



A.2.1.4.3 Fields

Field	Function
31 TX_IMAGE_FILTER_2_OVRD	TX Image Filter 2 Override Control If the TX_IMAGE_FILTER_OVRD_EN bit is set, then this bit turns on (=1) and off (=0) the Transmit Image Filter 2
30 TX_IMAGE_FILTER_1_OVRD	TX Image Filter 1 Override Control If the TX_IMAGE_FILTER_OVRD_EN bit is set, then this bit turns on (=1) and off (=0) the Transmit Image Filter 1
29 TX_IMAGE_FILTER_0_OVRD	TX Image Filter 0 Override Control If the TX_IMAGE_FILTER_OVRD_EN bit is set, then this bit turns on (=1) and off (=0) the Transmit Image Filter 0
28 TX_IMAGE_FILTER_OVRD_EN	TX Image Filter Override Enable This bit enables the TX Image Filter Override Control bits.
27 —	Reserved
26-24 GFSK_MOD_INDEX_SCALING	GFSK Modulation Index Scaling Factor This register selects the amount to scale the Transmitter Modulation Word in GFSK Protocols. Transmit modulation scaling adds the capability to scale the GFSK modulated output up or down by a factor of 1/32, 1/16 or 1/8. This is equivalent to having some additional programmability of the modulation index. 000b - 1 001b - 1 + 1/32

Table continues on the next page...

Field	Function
	010b - 1 + 1/16 011b - 1 + 1/8 100b - 1 - 1/32 101b - 1 - 1/16 110b - 1 - 1/8 111b - Reserved
23-22 —	Reserved
21 GFSK_FLD	Disable GFSK Filter Lookup Table 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.
20 GFSK_MLD	Disable GFSK Multiply Lookup Table If this bit is set, the GFSK Multiply Lookup table is disabled and GFSK_MULTIPLY_TABLE_MANUAL is used instead.
19-18 —	Reserved
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) } 00b - 0.32 01b - 0.50 10b - 0.70 11b - 1.00
15-0 GFSK_MULTIPLY_TABLE_MANUAL	Manual GFSK Multiply Lookup Table 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] }

A.2.1.5 TX GFSK Filter Coefficients 2 (GFSK_COEFF2)

A.2.1.5.1 Offset

Register	Offset
GFSK_COEFF2	Ch

A.2.1.5.2 Function

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 the GFSK Manual Filter Coefficient [63:0]

A.2.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GFSK_FILTER_COEFF_MANUAL2															
Reset	1	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GFSK_FILTER_COEFF_MANUAL2															
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1

A.2.1.5.4 Fields

Field	Function
31-0 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.

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

A.2.1.6 TX GFSK Filter Coefficients 1 (GFSK_COEFF1)

A.2.1.6.1 Offset

Register	Offset
GFSK_COEFF1	10h

A.2.1.6.2 Function

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 the GFSK Manual Filter Coefficient [63:0]

A.2.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GFSK_FILTER_COEFF_MANUAL1															
Reset	1	0	1	1	1	0	1	1	0	0	1	0	1	0	0	1

Bits	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	0	1	0	1	1	0	0	0	0	0	1	1	0	1

A.2.1.6.4 Fields

Field	Function
31-0 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.

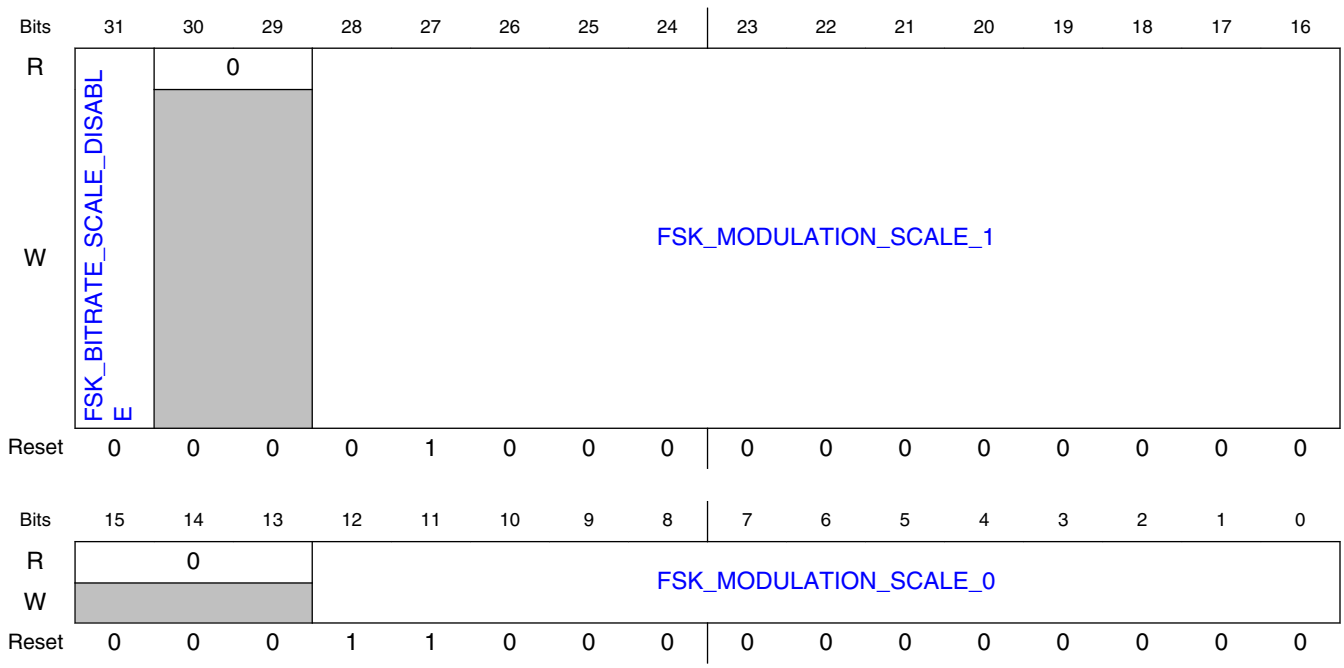
Bits	000 Filter Coefficients
[6:0]	Filter coeff 2 and 13
[15:7]	Filter coeff 6 and 9
[22:16]	Filter coeff 3 and 12
[31:23]	Filter coeff 7 and 8

A.2.1.7 TX FSK Modulation Levels (FSK_SCALE)

A.2.1.7.1 Offset

Register	Offset
FSK_SCALE	14h

A.2.1.7.2 Diagram



A.2.1.7.3 Fields

Field	Function
31 FSK_BITRATE_SCALE_DISABLE	FSK Bitrate Scaling Disable If this bit is set, the FSK Modulation values will not be scaled according to the bitrate setting.
30-29 —	Reserved
28-16 FSK_MODULATION_SCALE_1	FSK Modulation Scale for a data 1

Table continues on the next page...

Field	Function
	This register is used to provide the modulation level for a data 1 in FSK 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 range of signed values that this register supports is -4096 to 4095
15-13 —	Reserved
12-0 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 FSK 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 range of signed values that this register supports is -4096 to 4095

A.2.1.8 TX DFT Modulation Pattern (DFT_PATTERN)

A.2.1.8.1 Offset

Register	Offset
DFT_PATTERN	18h

A.2.1.8.2 Diagram

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

A.2.1.8.3 Fields

Field	Function
31-0	DFT Modulation Pattern

Field	Function
DFT_MOD_PATTERN	In TX DFT Pattern Register modes, if TX_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].

A.2.2 XCVR_PLL_DIG register descriptions

A.2.2.1 XCVR_PLL_DIG_ADDR Memory map

XCVR_PLL_DIG base address: 4005_C224h

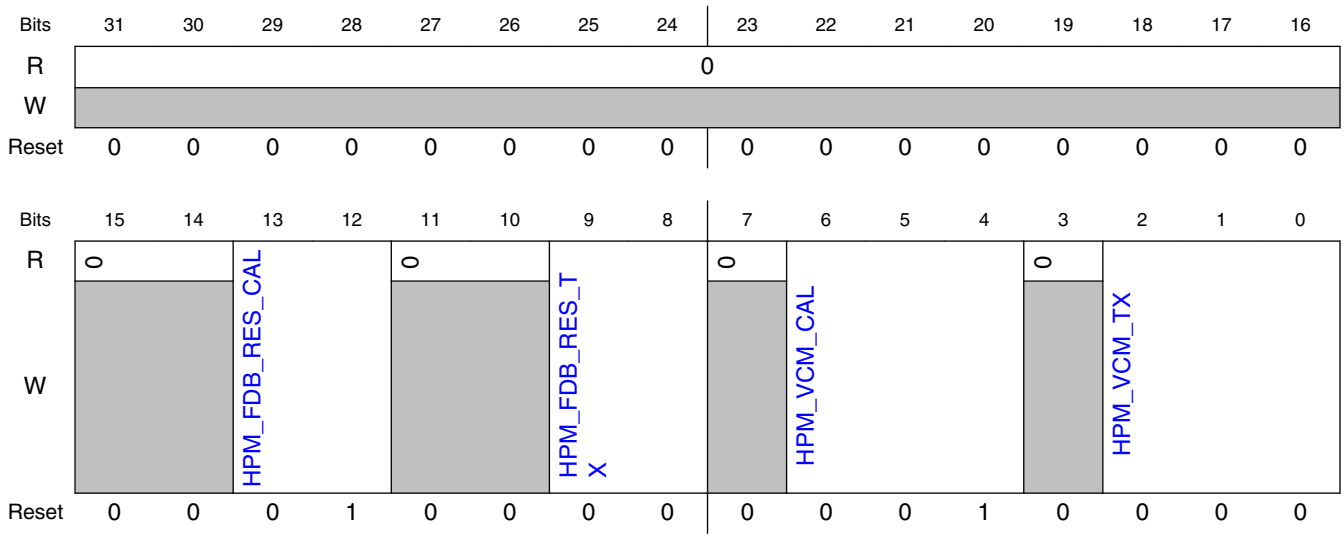
Offset	Register	Width (In bits)	Access	Reset value
0h	PLL HPM Analog Bump Control (HPM_BUMP)	32	RW	0000_1010h
4h	PLL Modulation Control (MOD_CTRL)	32	RW	0000_0000h
8h	PLL Channel Mapping (CHAN_MAP)	32	RW	0000_0200h
Ch	PLL Lock Detect Control (LOCK_DETECT)	32	RW	0060_6800h
10h	PLL High Port Modulator Control (HPM_CTRL)	32	RW	9084_0000h
20h	PLL High Port Sigma Delta Results (HPM_SDM_RES)	32	RW	0100_0000h
24h	PLL Low Port Modulator Control (LPM_CTRL)	32	RW	0202_0000h
28h	PLL Low Port Sigma Delta Control 1 (LPM_SDM_CTRL1)	32	RW	0026_0026h
2Ch	PLL Low Port Sigma Delta Control 2 (LPM_SDM_CTRL2)	32	RW	0008_0000h
30h	PLL Low Port Sigma Delta Control 3 (LPM_SDM_CTRL3)	32	RW	0010_0000h
34h	PLL Low Port Sigma Delta Result 1 (LPM_SDM_RES1)	32	RO	0000_0000h
38h	PLL Low Port Sigma Delta Result 2 (LPM_SDM_RES2)	32	RO	0400_0000h
3Ch	PLL Delay Matching (DELAY_MATCH)	32	RW	0000_0004h
40h	PLL Coarse Tune Control (CTUNE_CTRL)	32	RW	0000_0000h
54h	PLL Coarse Tune Results (CTUNE_RES)	32	RO	0962_0040h

A.2.2.2 PLL HPM Analog Bump Control (HPM_BUMP)

A.2.2.2.1 Offset

Register	Offset
HPM_BUMP	0h

A.2.2.2.2 Diagram



A.2.2.2.3 Fields

Field	Function
31-14 —	Reserved
13-12 HPM_FDB_RES_CAL	rfctrl_tx_dac_bump_fdb_res[1:0] during Calibration This is the value applied to the rfctrl_tx_dac_bump_fdb_res[1:0] port during High Port Calibration. This register sets the HPM DAC feedback resistor to increase the modulation gain during calibration. 00b - 29 kohms 01b - 58 kohms(gain of 2) 10b - 13 kohms 11b - 23.7 kohms
11-10 —	Reserved
9-8 HPM_FDB_RES_TX	rfctrl_tx_dac_bump_fdb_res[1:0] during Transmission This is the value applied to the rfctrl_tx_dac_bump_fdb_res[1:0] port during Radio Transmissions. This register sets the HPM DAC feedback resistor during transmissions. 00b - 29 kohms 01b - 58 kohms(gain of 2) 10b - 13 kohms 11b - 23.7 kohms
7 —	Reserved
6-4	rfctrl_tx_dac_bump_vcm[2:0] during Calibration This is the value applied to the rfctrl_tx_dac_bump_vcm[2:0] port during High Port Calibration.

Table continues on the next page...

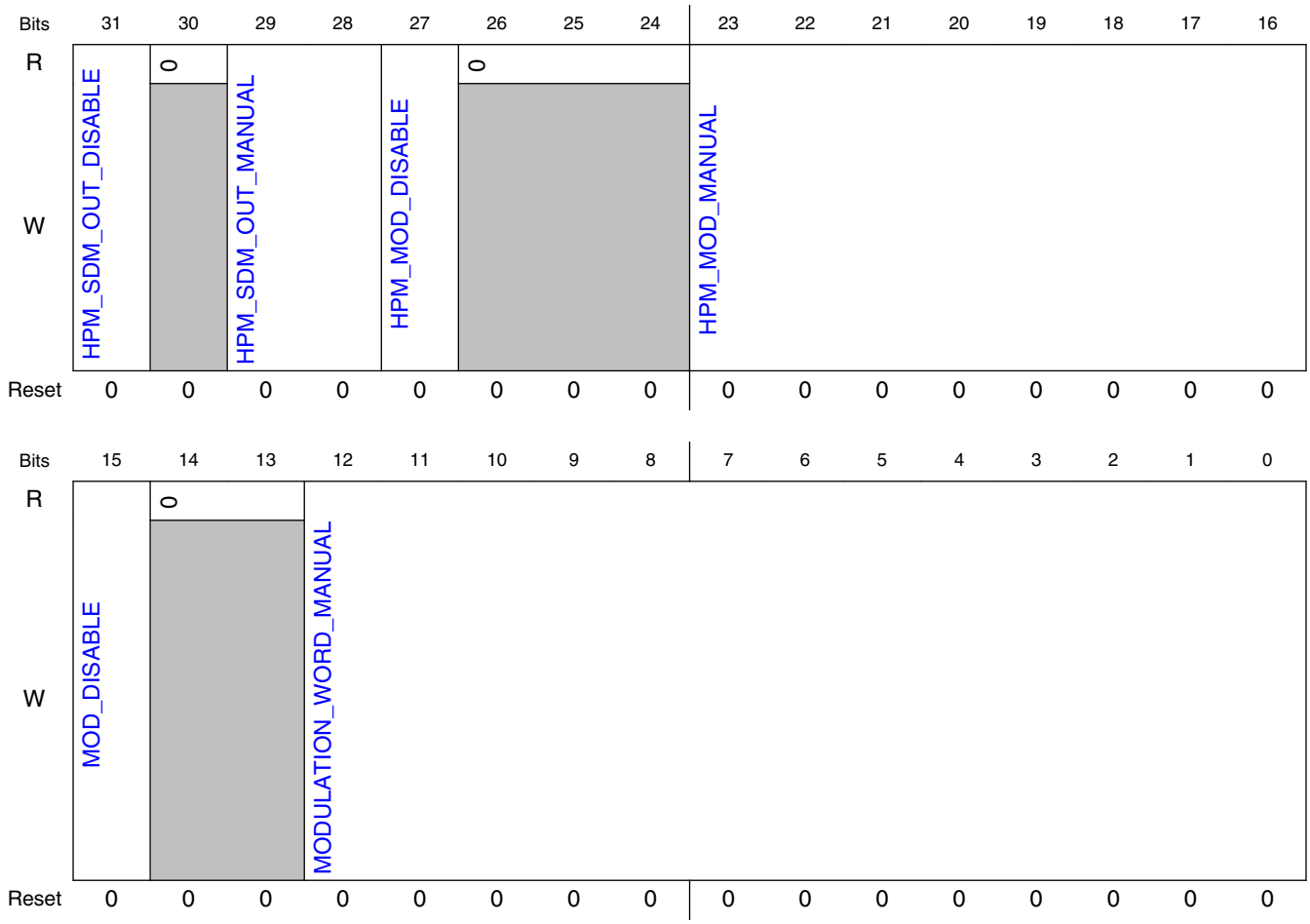
Field	Function
HPM_VCM_CAL	This register sets the HPM DAC Op-Amp reference voltage during calibration. 000b - 432 mV 001b - 328 mV 010b - 456 mV 011b - 473 mV 100b - 488 mV 101b - 408 mV 110b - 392 mV 111b - 376 mV
3 —	Reserved
2-0 HPM_VCM_TX	rfctrl_tx_dac_bump_vcm[2:0] during Transmission This is the value applied to the rfctrl_tx_dac_bump_vcm[2:0] port during Radio Transmissions. This register sets the HPM DAC Op-Amp reference voltage during transmissions. 000b - 432 mV 001b - 328 mV 010b - 456 mV 011b - 473 mV 100b - 488 mV 101b - 408 mV 110b - 392 mV 111b - 376 mV

A.2.2.3 PLL Modulation Control (MOD_CTRL)

A.2.2.3.1 Offset

Register	Offset
MOD_CTRL	4h

A.2.2.3.2 Diagram



A.2.2.3.3 Fields

Field	Function
31 HPM_SDM_OUT_DISABLE	Disable HPM SDM out If this bit is set, the High Port Sigma Delta Modulator output is disabled, and the High Port Fractional value applied to the VCO comes from the HPM_SDM_OUT_MANUAL register.
30 —	Reserved
29-28 HPM_SDM_OUT_MANUAL	Manual HPM SDM out If HPM_SDM_OUT_DISABLE is set, this register is the Fractional value that is applied to the VCO High Port.
27 HPM_MOD_DISABLE	Disable HPM Modulation If this bit is set, the High Port Modulation is disabled, and the High Port Modulation value applied to the VCO comes from the HPM_MOD_MANUAL register.
26-24 —	Reserved

Table continues on the next page...

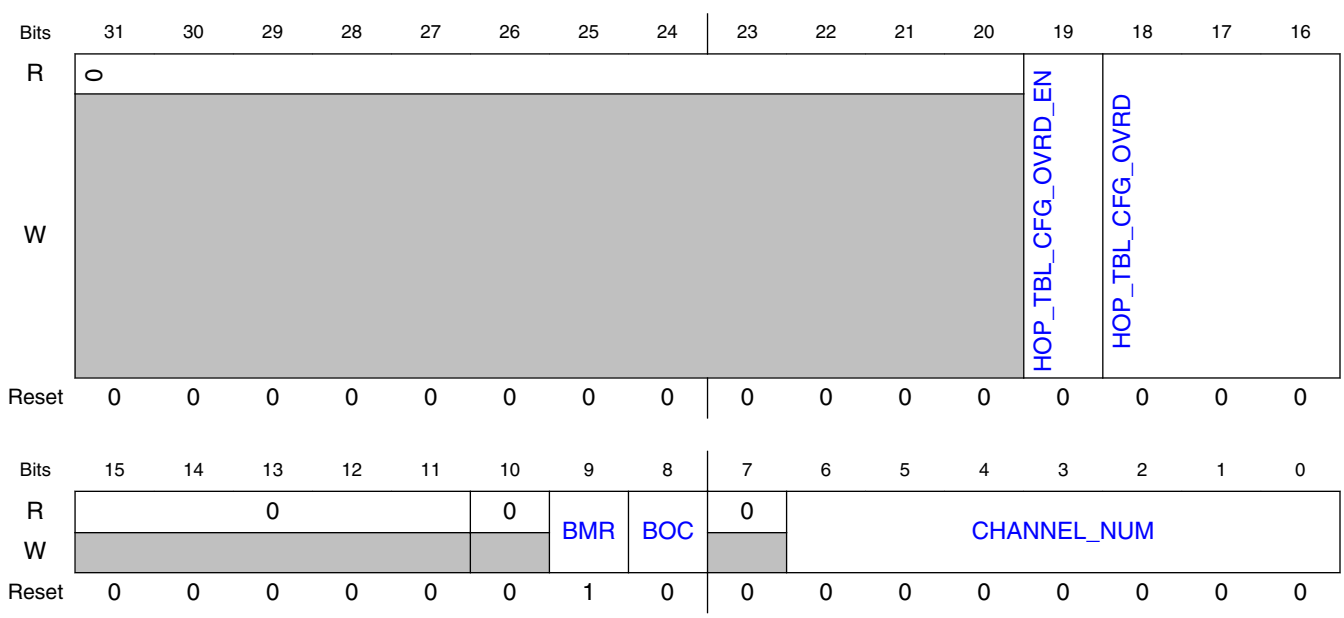
Field	Function
23-16 HPM_MOD_MANUAL	Manual HPM Modulation If HPM_MOD_DISABLE is set, this register is the modulation value that is applied to the VCO High Port.
15 MOD_DISABLE	Disable Modulation Word If this bit is set, any modulation from the TX Digital is disabled and the source of the Baseband Frequency Word is the MODULATION_WORD_MANUAL register.
14-13 —	Reserved
12-0 MODULATION_WORD_MANUAL	Manual Modulation Word If MOD_DISABLE is set, the signed 12 bit value that is represented by this register is the Baseband Frequency Word.

A.2.2.4 PLL Channel Mapping (CHAN_MAP)

A.2.2.4.1 Offset

Register	Offset
CHAN_MAP	8h

A.2.2.4.2 Diagram



A.2.2.4.3 Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31-20
—</td><td>Reserved</td></tr>
<tr><td>19
HOP_TBL_CFG_OVRD_EN</td><td>Hop Table Configuration Override Enable
If this bit is set then the HOP_TBL_CFG_OVRD and DFT_PATTERN Registers will be used to control the Coarse Tune, Channel Number, and Hop Table for the PLL Carrier Frequency Selection.</td></tr>
<tr><td>18-16
HOP_TBL_CFG_OVRD</td><td>Hop Table Configuration Override
If the HOP_TBL_CFG_OVRD_EN bit is set, then the DFT_PATTERN Register will be used to control the Hop Table for the PLL Carrier Frequency Selection.
010b - DFT_PATTERN[15:7] is signed offset to DFT_PATTERN[6:0] mapped channel number
011b - DFT_PATTERN[15:1] is signed Numerator, DFT_PATTERN[0] is integer selection</td></tr>
<tr><td>15-11
—</td><td>Reserved</td></tr>
<tr><td>10
—</td><td>Reserved</td></tr>
<tr><td>9
BMR</td><td>BLE MBAN Channel Remap
This bit controls the mapping of BLE channel 39 in Radio Protocol 2, BLE overlap MBAN mode.
0b - BLE channel 39 is mapped to BLE channel 39, 2.480 GHz
1b - BLE channel 39 is mapped to MBAN channel 39, 2.399 GHz</td></tr>
<tr><td>8
BOC</td><td>BLE Channel Number Override
This bit controls the source of the BLE channel selection.
0b - BLE channel number comes from the BLE Link Layer
1b - BLE channel number comes from the CHANNEL_NUM register (BLE protocols 0 and 2)</td></tr>
<tr><td>7
—</td><td>Reserved</td></tr>
<tr><td>6-0
CHANNEL_NUM</td><td>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 protocols 1, 6 and 7.
The Radio Channel Frequency can also be selected by setting the SDM_MAP_DISABLE bit in the PLL_LPM_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).
<table>
<tr><th colspan="6">Radio Protocols Supported, Channel Frequency in MHz</th></tr>
<tr><th colspan="2">BLE</th><th colspan="2">MBAN</th><th colspan="2">BLE/MBAN</th></tr>
<tr><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">2</td></tr>
<tr><td>Chan</td><td>MHz</td><td>Chan</td><td>MHz</td><td>Chan</td><td>MHz</td></tr>
</table>
</td></tr>
</table>

Field	Function					
	0	2,402	0	2,360	0	2,402
	1	2,404	1	2,361	1	2,404
	2	2,406	2	2,362	2	2,406
	3	2,408	3	2,363	3	2,408
	4	2,410	4	2,364	4	2,410
	5	2,412	5	2,365	5	2,412
	6	2,414	6	2,366	6	2,414
	7	2,416	7	2,367	7	2,416
	8	2,418	8	2,368	8	2,418
	9	2,420	9	2,369	9	2,420
	10	2,422	10	2,370	10	2,422
	11	2,424	11	2,371	11	2,424
	12	2,426	12	2,372	12	2,426
	13	2,428	13	2,373	13	2,428
	14	2,430	14	2,374	14	2,430
	15	2,432	15	2,375	15	2,432
	16	2,434	16	2,376	16	2,434
	17	2,436	17	2,377	17	2,436
	18	2,438	18	2,378	18	2,438
	19	2,440	19	2,379	19	2,440
	20	2,442	20	2,380	20	2,442
	21	2,444	21	2,381	21	2,444
	22	2,446	22	2,382	22	2,446
	23	2,448	23	2,383	23	2,448
	24	2,450	24	2,384	24	2,450
	25	2,452	25	2,385	25	2,452
	26	2,454	26	2,386	26	2,454
	27	2,456	27	2,387	27	2,456
	28	2,458	28	2,388	28	2,458
	29	2,460	29	2,389	29	2,460
	30	2,462	30	2,390	30	2,390
	31	2,464	31	2,391	31	2,391
	32	2,466	32	2,392	32	2,392
	33	2,468	33	2,393	33	2,393
	34	2,470	34	2,394	34	2,394
	35	2,472	35	2,395	35	2,395
	36	2,474	36	2,396	36	2,396
	37	2,476	37	2,397	37	2,397
	38	2,478	38	2,398	38	2,398
	39	2,480	39	2,399	39*	2,480

Field	Function

* The BLE MBAN Channel Remap bit, BMR, controls the frequency mapping in this case.

Radio Protocols Supported, Channel Frequency in MHz					
DFT, Generic					
6,7,8,9					
Channel	MHz	Channel	MHz	Channel	MHz
0	2,360	43	2,403	86	2,446
1	2,361	44	2,404	87	2,447
2	2,362	45	2,405	88	2,448
3	2,363	46	2,406	89	2,449
4	2,364	47	2,407	90	2,450
5	2,365	48	2,408	91	2,451
6	2,366	49	2,409	92	2,452
7	2,367	50	2,410	93	2,453
8	2,368	51	2,411	94	2,454
9	2,369	52	2,412	95	2,455
10	2,370	53	2,413	96	2,456
11	2,371	54	2,414	97	2,457
12	2,372	55	2,415	98	2,458
13	2,373	56	2,416	99	2,459
14	2,374	57	2,417	100	2,460
15	2,375	58	2,418	101	2,461
16	2,376	59	2,419	102	2,462
17	2,377	60	2,420	103	2,463
18	2,378	61	2,421	104	2,464
19	2,379	62	2,422	105	2,465
20	2,380	63	2,423	106	2,466
21	2,381	64	2,424	107	2,467
22	2,382	65	2,425	108	2,468
23	2,383	66	2,426	109	2,469
24	2,384	67	2,427	110	2,470
25	2,385	68	2,428	111	2,471
26	2,386	69	2,429	112	2,472
27	2,387	70	2,430	113	2,473
28	2,388	71	2,431	114	2,474
29	2,389	72	2,432	115	2,475
30	2,390	73	2,433	116	2,476
31	2,391	74	2,434	117	2,477
32	2,392	75	2,435	118	2,478
33	2,393	76	2,436	119	2,479
34	2,394	77	2,437	120	2,480

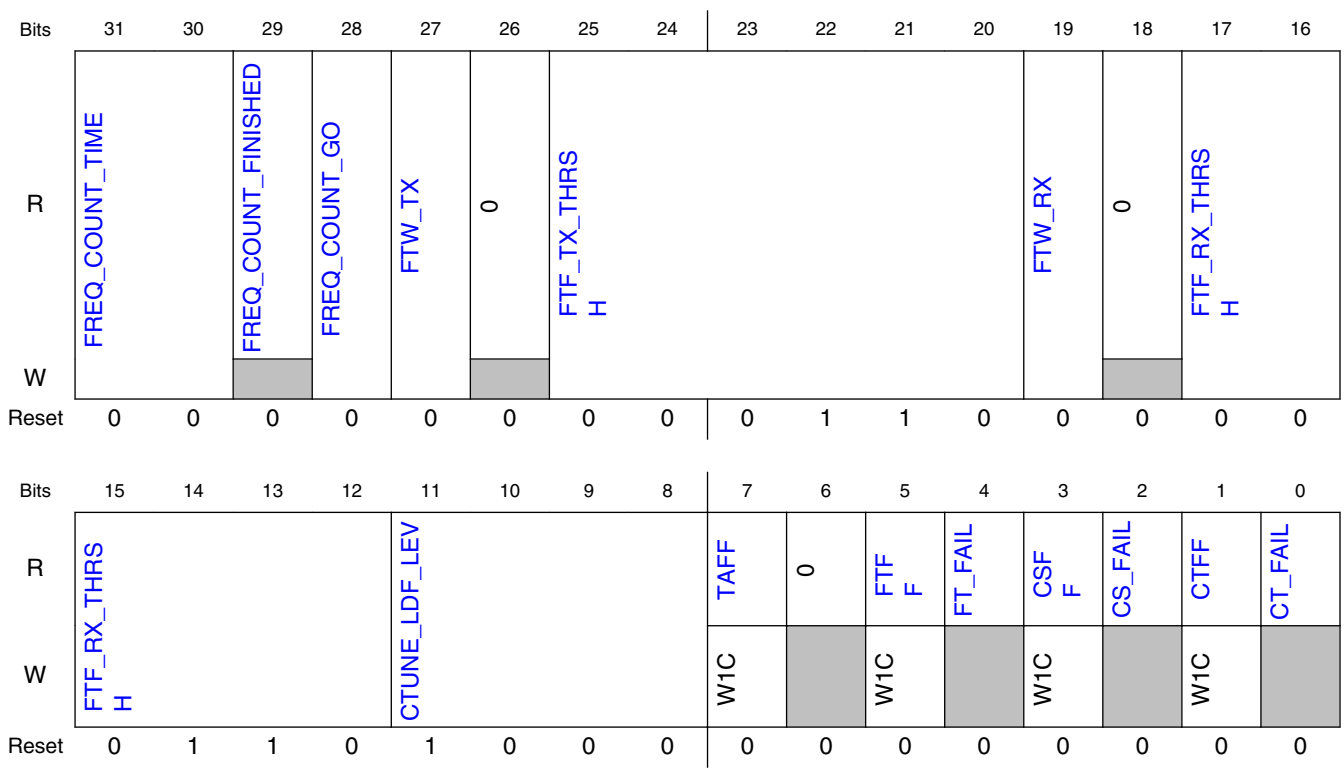
Field	Function					
	35	2,395	78	2,438	121	2,481
	36	2,396	79	2,439	122	2,482
	37	2,397	80	2,440	123	2,483
	38	2,398	81	2,441	124	2,484
	39	2,399	82	2,442	125	2,485
	40	2,400	83	2,443	126	2,486
	41	2,401	84	2,444	127	2,487
	42	2,402	85	2,445		

A.2.2.5 PLL Lock Detect Control (LOCK_DETECT)

A.2.2.5.1 Offset

Register	Offset
LOCK_DETECT	Ch

A.2.2.5.2 Diagram



A.2.2.5.3 Fields

Field	Function
31-30 FREQ_COUNT_TIME	Frequency Meter Count Time This is the length of time that the Frequency Meter will count VCO clocks. 00b - 800 us 01b - 25 us 10b - 50 us 11b - 100 us
29 FREQ_COUNT_FINISHED	Frequency Meter has finished the Count Time This bit is set when the FREQ_COUNT_TIME value is reached, it is cleared when FREQ_COUNT_GO is cleared.
28 FREQ_COUNT_GO	Start the Frequency Meter The Frequency Meter starts when this bit is set and runs until the FREQ_COUNT_TIME value is reached. The bit should not be cleared until after the counting is finished. After the Counting Time finishes, the FREQ_COUNT_FINISHED bit will read as a one. Then the measured frequency can be calculated by reading the frequency counts in the DFT_FREQ_COUNTER register (in the XCVR_ANALOG register space) and dividing by the FREQ_COUNT_TIME. Note that the counting is done at the VCO frequency which is 2X the Carrier Frequency
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. 0b - 4 us 1b - 8 us
26 —	Reserved
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. 0b - 4 us 1b - 8 us
18 —	Reserved
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.

Table continues on the next page...

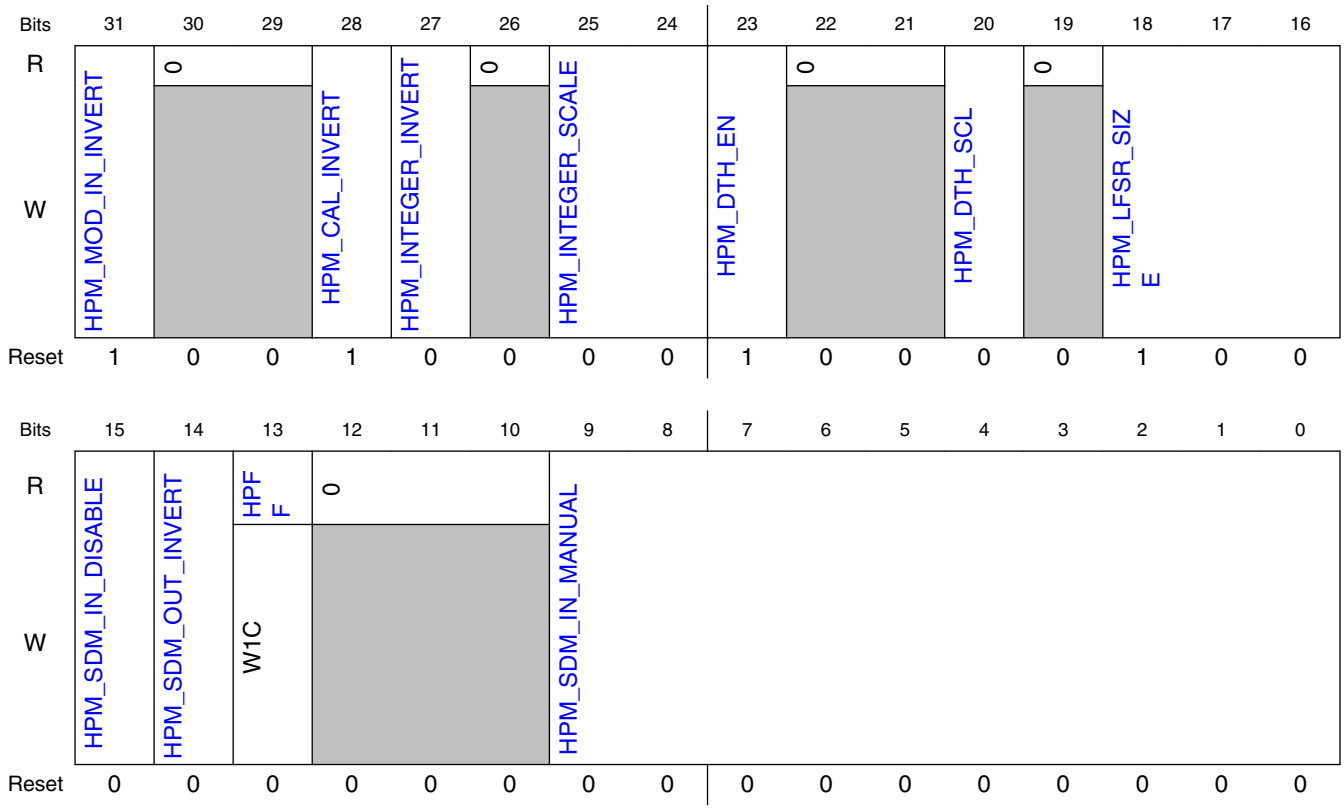
Field	Function
7 TAFF	TSM Abort Failure Flag This bit is set if the TSM Sequence Aborts, and this bit is cleared by writing a 1 to it.
6 —	Reserved
5 FTFF	Frequency Target Failure Flag This bit is set when FT_FAIL is first set, and this bit is cleared by writing a 1 to it.
4 FT_FAIL	Real time status 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.
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.

A.2.2.6 PLL High Port Modulator Control (HPM_CTRL)

A.2.2.6.1 Offset

Register	Offset
HPM_CTRL	10h

A.2.2.6.2 Diagram



A.2.2.6.3 Fields

Field	Function
31 HPM_MOD_IN_INVERT	Invert High Port Modulation If this bit is set then the High Port Modulation Word is inverted before it is multiplied and split into Integer and Fractional values.
30-29 —	Reserved
28 HPM_CAL_INVERT	Invert High Port Modulator Calibration If this bit is set then the order of the High Port Calibration is reversed in order to get a positive count difference between the Maximum and Minimum settings of the HPM DAC.
27 HPM_INTEGER_INVERT	Invert High Port Modulation Integer If this bit is set then the High Port Modulation Integer value, after the multiply and split into Integer and Fractional values, will be inverted before it is applied to the VCO High Port DAC Array.
26 —	Reserved
25-24 HPM_INTEGER_SCALE	High Port Modulation Integer Scale This register controls the scaling of the High Port Modulation Integer Value applied to the VCO High Port DAC Array.

Table continues on the next page...

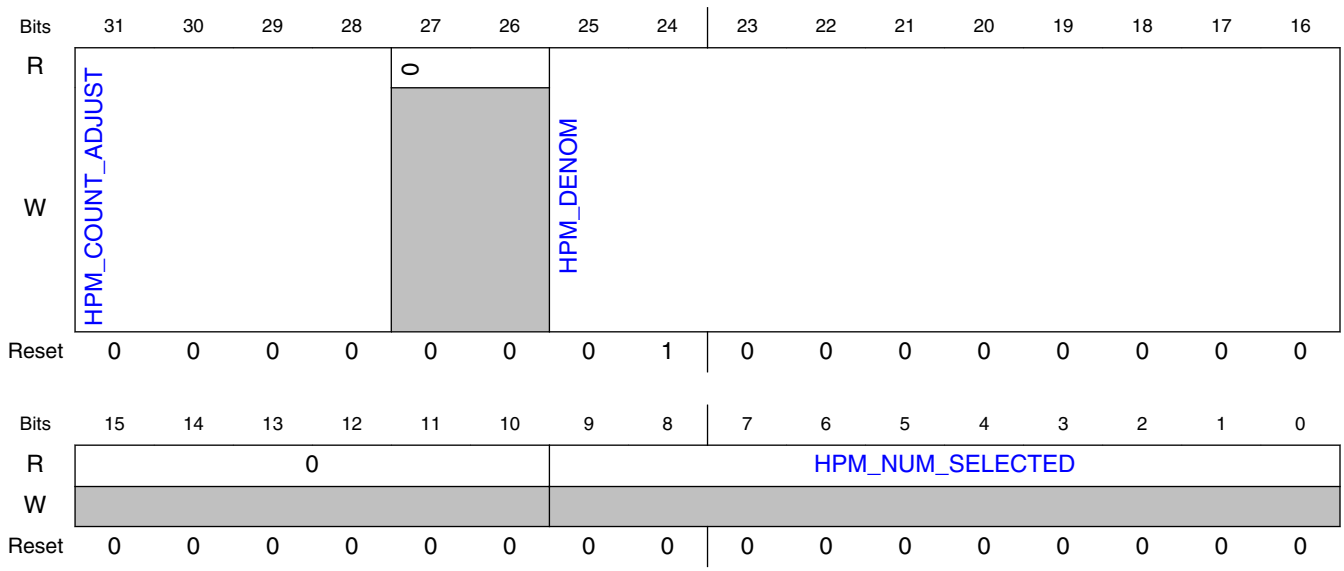
Field	Function
	00b - No Scaling 01b - Multiply by 2 10b - Divide by 2 11b - Reserved
23 HPM_DTH_EN	Dither Enable for HPM LFSR If this bit is set, the High Port Fraction will be Dithered by the High Port LFSR before it is applied to the High Port SDM.
22-21 —	Reserved
20 HPM_DTH_SCL	HPM Dither Scale If this bit is set, the LFSR dithering of the High Port Fraction will be multiplied by 2.
19 —	Reserved
18-16 HPM_LFSR_SIZE	HPM LFSR Length This register selects the length of the HPM LFSR and the associated LFSR Tap Mask 000b - LFSR 9, tap mask 100010000 001b - LFSR 10, tap mask 1001000000 010b - LFSR 11, tap mask 11101000000 011b - LFSR 13, tap mask 1101100000000 100b - LFSR 15, tap mask 111010000000000 101b - LFSR 17, tap mask 11110000000000000 110b - Reserved 111b - Reserved
15 HPM_SDM_IN_DISABLE	Disable HPM SDM Input If this bit is set, the Fractional portion of the High Port Modulation to the High Port SDM is disabled, and the High Port SDM input comes from the HPM_SDM_IN_MANUAL register.
14 HPM_SDM_OUT_INVERT	Invert HPM SDM Output If this bit is set the High Port SDM result will be Inverted before it is applied to the VCO High Port DAC Array.
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
9-0 HPM_SDM_IN_MANUAL	Manual High Port SDM Fractional value If HPM_SDM_IN_DISABLE is set, this register is the value that is applied as the Fractional value to the input of the High Port SDM.

A.2.2.7 PLL High Port Sigma Delta Results (HPM_SDM_RES)

A.2.2.7.1 Offset

Register	Offset
HPM_SDM_RES	20h

A.2.2.7.2 Diagram



A.2.2.7.3 Fields

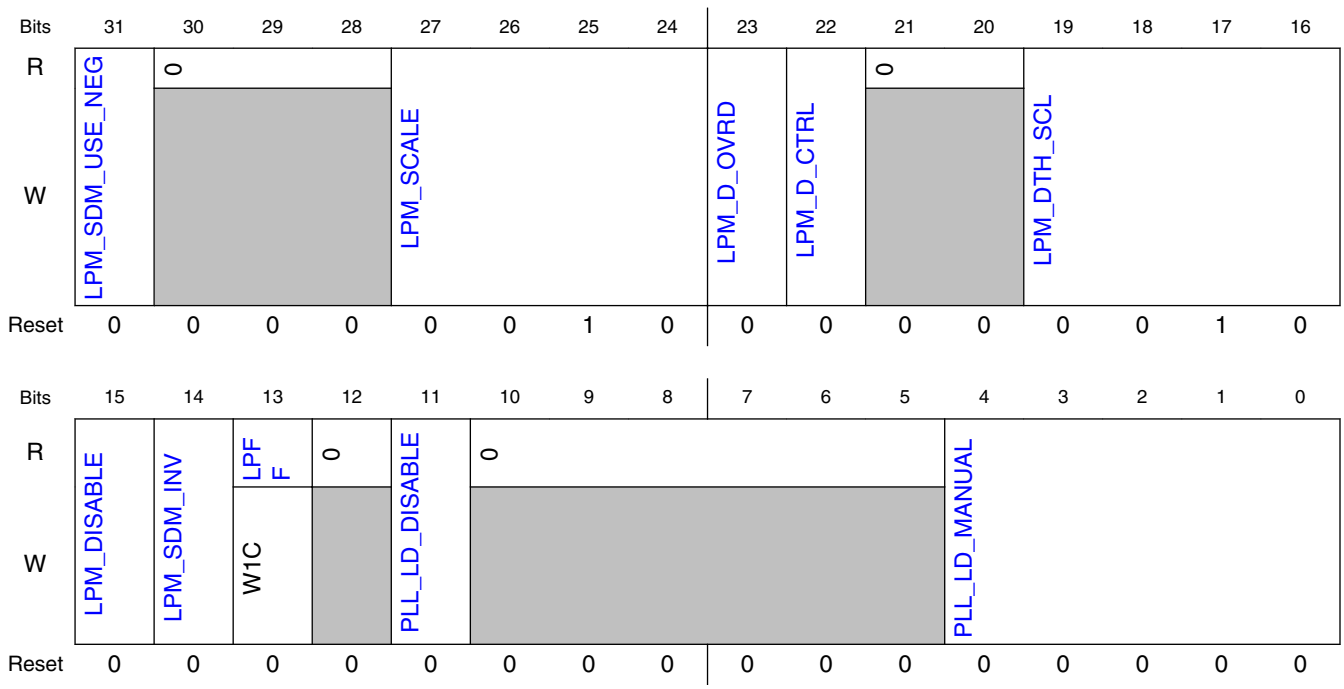
Field	Function
31-28 HPM_COUNT_ADJUST	HPM_COUNT_ADJUST This register represents a signed three bit value that is used to adjust the High Port Calibration Frequency Count Difference. The range of adjustment is -8 to +7, applied to the difference between Count 1 and Count 2.
27-26 —	Reserved
25-16 HPM_DENOM	High Port Modulator SDM 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
15-10 —	Reserved
9-0 HPM_NUM_SELECTED	High Port Modulator SDM Numerator 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

A.2.2.8 PLL Low Port Modulator Control (LPM_CTRL)

A.2.2.8.1 Offset

Register	Offset
LPM_CTRL	24h

A.2.2.8.2 Diagram



A.2.2.8.3 Fields

Field	Function
31 LPM_SDM_USE_NEG	Use the Negedge of the Sigma Delta clock If this bit is set then the negative edge of the Sigma Delta clock is used to launch the Low Port Modulation word to the VCO Loop Divider.
30-28 —	Reserved
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.

Table continues on the next page...

Field	Function
	0000b - No Scaling 0001b - Multiply by 2 0010b - Multiply by 4 0011b - Multiply by 8 0100b - Multiply by 16 0101b - Multiply by 32 0110b - Multiply by 64 0111b - Multiply by 128 1000b - Multiply by 256, this is the intended setting for normal operation. 1001b - Multiply by 512 1010b - Multiply by 1024 1011b - Multiply by 2048 1100b - Reserved 1101b - Reserved 1110b - Reserved 1111b - 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 in Radio Transmit mode 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
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. 0000b - Reserved 0001b - Reserved 0010b - Reserved 0011b - Reserved 0100b - Reserved 0101b - -128 to 96 0110b - -256 to 192 0111b - -512 to 384 1000b - -1024 to 768, this is the intended setting for normal operation. 1001b - -2048 to 1536 1010b - -4096 to 3072 1011b - -8192 to 6144 1100b - Reserved 1101b - Reserved 1110b - Reserved 1111b - Reserved
15 LPM_DISABLE	Disable LPM SDM This bit controls the Modulation of the Low Port Sigma Delta. If this bit is set, the Low Port Sigma Delta Modulator 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.

Table continues on the next page...

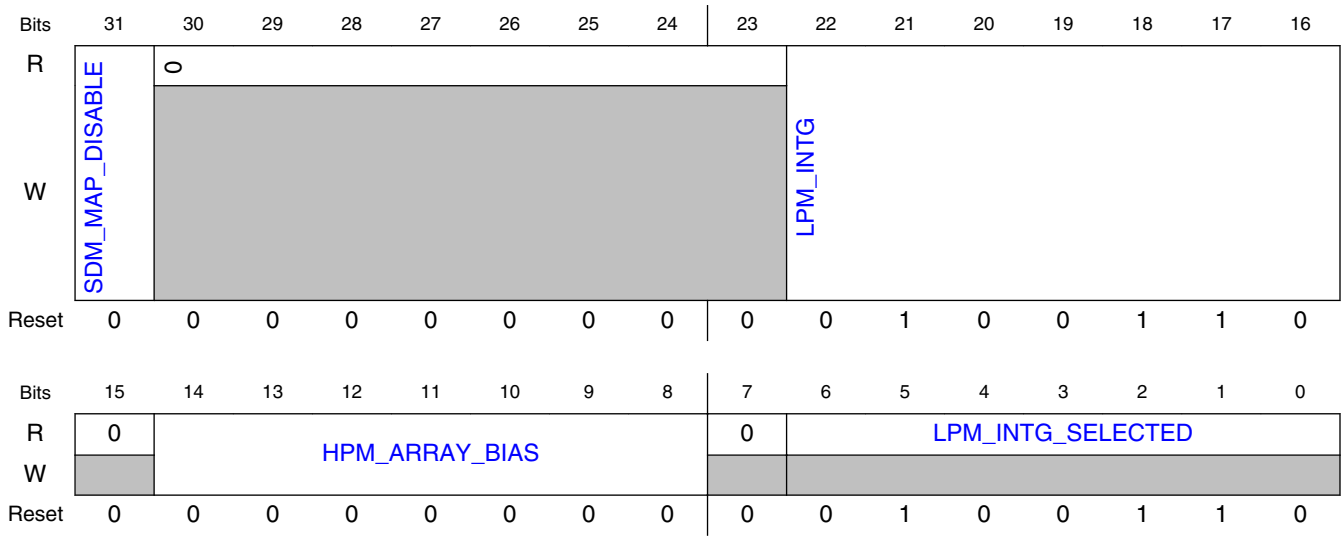
Field	Function
	No Modulation or Dithering will be added to the steady frequency result.
14 LPM_SDM_INV	Invert LPM SDM If this bit is set the Scaled Baseband Frequency Word, including any Dithering, will be Inverted before it is applied to the Low Port Sigma Delta Modulator.
13 LPFF	LPM SDM Invalid Flag 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
11 PLL_LD_DISABLE	Disable PLL Loop Divider 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_LD_MANUAL register.
10-5 —	Reserved
4-0 PLL_LD_MANUAL	Manual PLL Loop Divider value If PLL_LD_DISABLE is set, this register is the value that is applied to the PLL Loop Divider.

A.2.2.9 PLL Low Port Sigma Delta Control 1 (LPM_SDM_CTRL1)

A.2.2.9.1 Offset

Register	Offset
LPM_SDM_CTRL1	28h

A.2.2.9.2 Diagram



A.2.2.9.3 Fields

Field	Function
31 SDM_MAP_DISABLE	Disable SDM Mapping 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
22-16 LPM_INTG	Manual Low Port Modulation Integer Value If SDM_MAP_DISABLE 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 —	Reserved
14-8 HPM_ARRAY_BIAS	Bias value for High Port DAC Array Midpoint The value of this register represents a signed six bit value that can be used to adjust the midpoint of the High Port Modulator DAC. The range of adjustment is -64 to +63, and the adjustment is made to the High Port Modulation Word before the multiply, and before the split into Integer and Fractional values. The range of adjustment in Hz is approximately +/- 62.5 kHz
7 —	Reserved
6-0 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.

A.2.2.10 PLL Low Port Sigma Delta Control 2 (LPM_SDM_CTRL2)

A.2.2.10.1 Offset

Register	Offset
LPM_SDM_CTRL2	2Ch

A.2.2.10.2 Diagram

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

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

A.2.2.10.3 Fields

Field	Function
31-28 —	Reserved
27-0 LPM_NUM	Low Port Modulation Numerator If SDM_MAP_DISABLE 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

A.2.2.11 PLL Low Port Sigma Delta Control 3 (LPM_SDM_CTRL3)

A.2.2.11.1 Offset

Register	Offset
LPM_SDM_CTRL3	30h

A.2.2.11.2 Diagram

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

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

A.2.2.11.3 Fields

Field	Function
31-28 —	Reserved
27-0 LPM_DENOM	Low Port Modulation Denominator If SDM_MAP_DISABLE 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

A.2.2.12 PLL Low Port Sigma Delta Result 1 (LPM_SDM_RES1)

A.2.2.12.1 Offset

Register	Offset
LPM_SDM_RES1	34h

A.2.2.12.2 Diagram

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

A.2.2.12.3 Fields

Field	Function
31-28 —	Reserved
27-0 LPM_NUM_SELECTED	Low Port Modulation Numerator Applied This is the value that is currently being applied to the Low Port Sigma Delta Modulator, for valid Sigma Delta operation the resulting fraction NUM/DENOM must be in the range -0.6 to +0.6

A.2.2.13 PLL Low Port Sigma Delta Result 2 (LPM_SDM_RES2)

A.2.2.13.1 Offset

Register	Offset
LPM_SDM_RES2	38h

A.2.2.13.2 Diagram

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

A.2.2.13.3 Fields

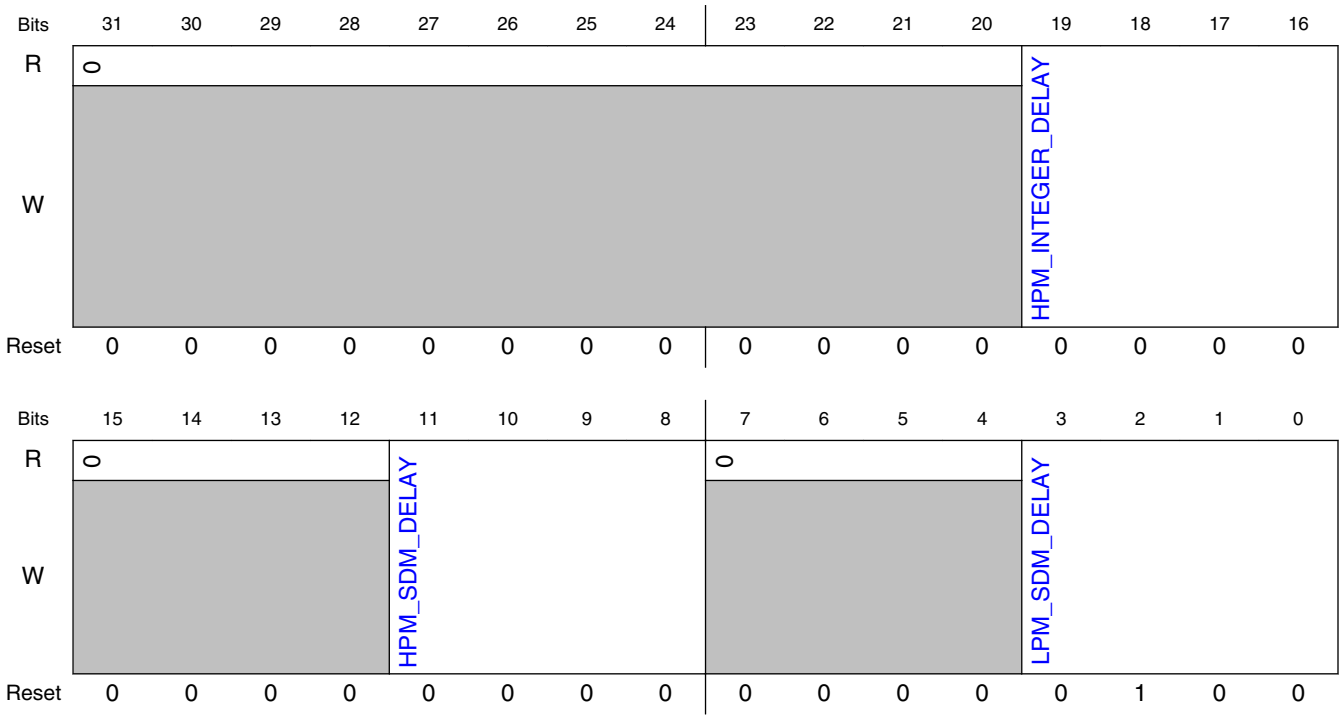
Field	Function
31-28 —	Reserved
27-0 LPM_DENOM_SELECTED	Low Port Modulation Denominator Selected This is the value that is currently being applied to the Low Port Sigma Delta Modulator, for valid Sigma Delta operation the resulting fraction NUM/DENOM must be in the range -0.6 to +0.6

A.2.2.14 PLL Delay Matching (DELAY_MATCH)

A.2.2.14.1 Offset

Register	Offset
DELAY_MATCH	3Ch

A.2.2.14.2 Diagram



A.2.2.14.3 Fields

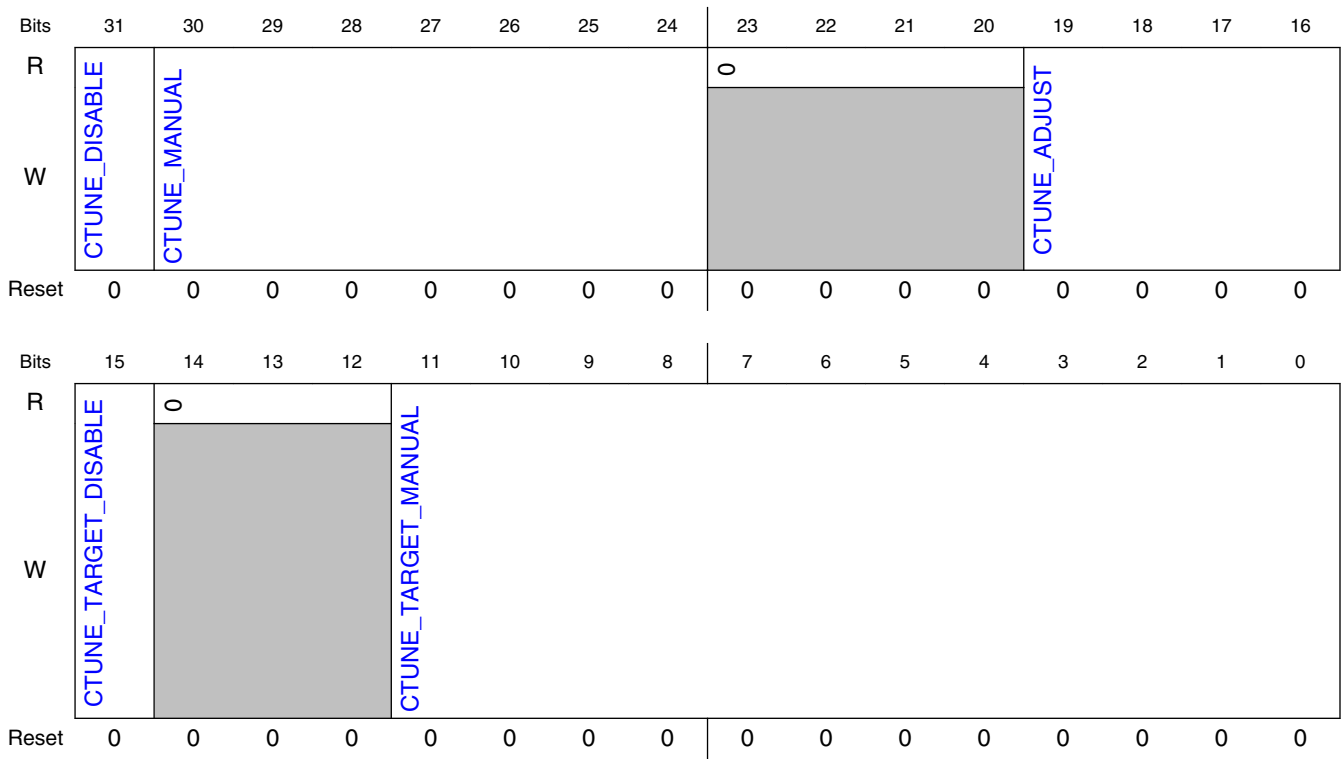
Field	Function
31-20 —	Reserved
19-16 HPM_INTEGER_DELAY	High Port Integer Delay Matching This register selects the number of clock cycles of the PLL Sigma Delta Clock divided by 2 to delay the High Port Integer modulation of the VCO High Port DAC Array.
15-12 —	Reserved
11-8 HPM_SDM_DELAY	High Port SDM Delay Matching 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 Fraction. 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
3-0 LPM_SDM_DELAY	Low Port SDM Delay Matching 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.

A.2.2.15 PLL Coarse Tune Control (CTUNE_CTRL)

A.2.2.15.1 Offset

Register	Offset
CTUNE_CTRL	40h

A.2.2.15.2 Diagram



A.2.2.15.3 Fields

Field	Function
31 CTUNE_DISABLE	Coarse Tune Disable If this bit is set, the Coarse Tune Setting applied to the VCO comes from the CTUNE_MANUAL register.
30-24 CTUNE_MANUAL	Manual Coarse Tune Setting If CTUNE_DISABLE is set, this register is the value that is applied to the VCO as the Coarse Tune Setting.
23-20 —	Reserved

Table continues on the next page...

Field	Function
19-16 CTUNE_ADJUST	Coarse Tune Count Adjustment This register is a signed three bit value that adjusts the PLL Frequency Meter count used in the Coarse Tune Calibration. The range of adjustment is -8 to +7, and the adjustment is only made to the PLL Frequency count used by the Coarse Tune Calibration sequence.
15 CTUNE_TARGET_DISABLE	Disable Coarse Tune Target If this bit is set, the Frequency Target presented to the Coarse Tune Calibrator comes from the CTUNE_TARGET_MANUAL register.
14-12 —	Reserved
11-0 CTUNE_TARGET_MANUAL	Manual Coarse Tune Target 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.

A.2.2.16 PLL Coarse Tune Results (CTUNE_RES)

A.2.2.16.1 Offset

Register	Offset
CTUNE_RES	54h

A.2.2.16.2 Diagram

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

Bits	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

A.2.2.16.3 Fields

Field	Function
31-28	Reserved

Table continues on the next page...

Field	Function
—	
27-16 CTUNE_FREQ_SELECTED	Coarse Tune Frequency Selected 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
6-0 CTUNE_SELECTED	Coarse Tune Setting to VCO This is the current VCO Coarse Tune setting, it is the result of the Coarse Tune Calibration, unless overridden using CTUNE_DISABLE.

A.2.3 XCVR_RX_DIG register descriptions

A.2.3.1 XCVR_RX_DIG_ADDR Memory map

XCVR_RX_DIG base address: 4005_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	RX Digital Control (RX_DIG_CTRL)	32	RW	0000_0000h
4h	AGC Control 0 (AGC_CTRL_0)	32	RW	0000_0000h
8h	AGC Control 1 (AGC_CTRL_1)	32	RW	0000_0000h
Ch	AGC Control 2 (AGC_CTRL_2)	32	RW	00A6_9000h
10h	AGC Control 3 (AGC_CTRL_3)	32	RW	0000_0000h
14h	AGC Status (AGC_STAT)	32	RO	0000_0000h
18h	RSSI Control 0 (RSSI_CTRL_0)	32	RW	0030_0000h
1Ch	RSSI Control 1 (RSSI_CTRL_1)	32	RO	0000_0000h
24h	DCOC Control 0 (DCOC_CTRL_0)	32	RW	0000_0000h
28h	DCOC Control 1 (DCOC_CTRL_1)	32	RW	0000_0000h
2Ch	DCOC DAC Initialization (DCOC_DAC_INIT)	32	RW	8080_2020h
30h	DCOC Digital Correction Manual Override (DCOC_DIG_MAN)	32	RW	0000_0000h
34h	DCOC Calibration Gain (DCOC_CAL_GAIN)	32	RW	0000_0000h
38h	DCOC Status (DCOC_STAT)	32	RO	8080_2020h
3Ch	DCOC DC Estimate (DCOC_DC_EST)	32	RO	0000_0000h
40h	DCOC Calibration Reciprocals (DCOC_CAL_RCP)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
48h	IQMC Control (IQMC_CTRL)	32	RW	0400_8000h
4Ch	IQMC Calibration (IQMC_CAL)	32	RW	0000_0400h
50h	LNA_GAIN Step Values 3..0 (LNA_GAIN_VAL_3_0)	32	RW	3809_321Dh
54h	LNA_GAIN Step Values 7..4 (LNA_GAIN_VAL_7_4)	32	RW	8B74_5D4Fh
58h	LNA_GAIN Step Values 8 (LNA_GAIN_VAL_8)	32	RW	0000_B6A1h
5Ch	BBA Resistor Tune Values 7..0 (BBA_RES_TUNE_VAL_7_0)	32	RW	0000_0000h
60h	BBA Resistor Tune Values 10..8 (BBA_RES_TUNE_VAL_10_8)	32	RW	0000_0000h
64h	LNA Linear Gain Values 2..0 (LNA_GAIN_LIN_VAL_2_0)	32	RW	0000_0000h
68h	LNA Linear Gain Values 5..3 (LNA_GAIN_LIN_VAL_5_3)	32	RW	0000_0000h
6Ch	LNA Linear Gain Values 8..6 (LNA_GAIN_LIN_VAL_8_6)	32	RW	0000_0000h
70h	LNA Linear Gain Values 9 (LNA_GAIN_LIN_VAL_9)	32	RW	0000_0000h
74h	BBA Resistor Tune Values 3..0 (BBA_RES_TUNE_LIN_VAL_3_0)	32	RW	0000_0000h
78h	BBA Resistor Tune Values 7..4 (BBA_RES_TUNE_LIN_VAL_7_4)	32	RW	0000_0000h
7Ch	BBA Resistor Tune Values 10..8 (BBA_RES_TUNE_LIN_VAL_10_8)	32	RW	0000_0000h
80h	AGC Gain Tables Step 03..00 (AGC_GAIN_TBL_03_00)	32	RW	0000_0000h
84h	AGC Gain Tables Step 07..04 (AGC_GAIN_TBL_07_04)	32	RW	0000_0000h
88h	AGC Gain Tables Step 11..08 (AGC_GAIN_TBL_11_08)	32	RW	0000_0000h
8Ch	AGC Gain Tables Step 15..12 (AGC_GAIN_TBL_15_12)	32	RW	0000_0000h
90h	AGC Gain Tables Step 19..16 (AGC_GAIN_TBL_19_16)	32	RW	0000_0000h
94h	AGC Gain Tables Step 23..20 (AGC_GAIN_TBL_23_20)	32	RW	0000_0000h
98h	AGC Gain Tables Step 26..24 (AGC_GAIN_TBL_26_24)	32	RW	0000_0000h
A0h - 108h	DCOC Offset (DCOC_OFFSET_0 - DCOC_OFFSET_26)	32	RW	0000_0000h
10Ch	DCOC BBA DAC Step (DCOC_BBA_STEP)	32	RW	0000_0000h
110h	DCOC TZA DAC Step 0 (DCOC_TZA_STEP_0)	32	RW	0000_0000h
114h	DCOC TZA DAC Step 1 (DCOC_TZA_STEP_1)	32	RW	0000_0000h
118h	DCOC TZA DAC Step 2 (DCOC_TZA_STEP_2)	32	RW	0000_0000h
11Ch	DCOC TZA DAC Step 3 (DCOC_TZA_STEP_3)	32	RW	0000_0000h
120h	DCOC TZA DAC Step 4 (DCOC_TZA_STEP_4)	32	RW	0000_0000h
124h	DCOC TZA DAC Step 5 (DCOC_TZA_STEP_5)	32	RW	0000_0000h
128h	DCOC TZA DAC Step 6 (DCOC_TZA_STEP_6)	32	RW	0000_0000h
12Ch	DCOC TZA DAC Step 7 (DCOC_TZA_STEP_7)	32	RW	0000_0000h
130h	DCOC TZA DAC Step 5 (DCOC_TZA_STEP_8)	32	RW	0000_0000h
134h	DCOC TZA DAC Step 9 (DCOC_TZA_STEP_9)	32	RW	0000_0000h
138h	DCOC TZA DAC Step 10 (DCOC_TZA_STEP_10)	32	RW	0000_0000h
160h	DCOC Calibration Fail Thresholds (DCOC_CAL_FAIL_TH)	32	RW	0000_0000h
164h	DCOC Calibration Pass Thresholds (DCOC_CAL_PASS_TH)	32	RW	0000_0000h
168h	DCOC Calibration Alpha (DCOC_CAL_ALPHA)	32	RO	0000_0000h
16Ch	DCOC Calibration Beta Q (DCOC_CAL_BETA_Q)	32	RO	0000_0000h
170h	DCOC Calibration Beta I (DCOC_CAL_BETA_I)	32	RO	0000_0000h
174h	DCOC Calibration Gamma (DCOC_CAL_GAMMA)	32	RO	0000_0000h

Table continues on the next page...

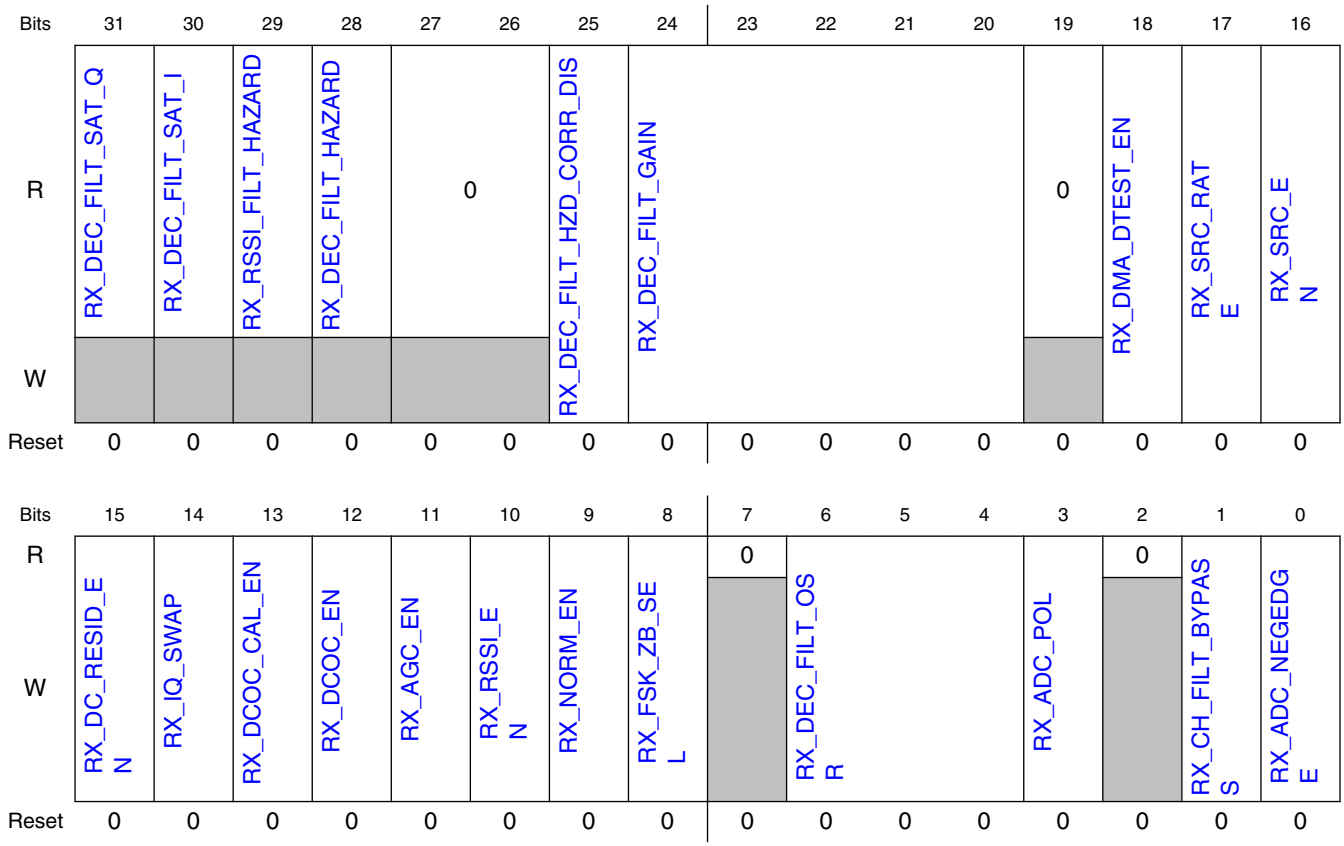
Offset	Register	Width (In bits)	Access	Reset value
178h	DCOC Calibration IIR (DCOC_CAL_IIR)	32	RW	0000_0000h
180h - 188h	DCOC Calibration Result (DCOC_CAL1 - DCOC_CAL3)	32	RO	0000_0000h
190h	RX_DIG CCA ED LQI Control Register 0 (CCA_ED_LQI_CTRL_0)	32	RW	0000_0000h
194h	RX_DIG CCA ED LQI Control Register 1 (CCA_ED_LQI_CTRL_1)	32	RW	0000_0000h
198h	RX_DIG CCA ED LQI Status Register 0 (CCA_ED_LQI_STAT_0)	32	RO	0000_0000h
1A0h	Receive Channel Filter Coefficient 0 (RX_CHF_COEF_0)	32	RW	0000_0000h
1A4h	Receive Channel Filter Coefficient 1 (RX_CHF_COEF_1)	32	RW	0000_0000h
1A8h	Receive Channel Filter Coefficient 2 (RX_CHF_COEF_2)	32	RW	0000_0000h
1ACh	Receive Channel Filter Coefficient 3 (RX_CHF_COEF_3)	32	RW	0000_0000h
1B0h	Receive Channel Filter Coefficient 4 (RX_CHF_COEF_4)	32	RW	0000_0000h
1B4h	Receive Channel Filter Coefficient 5 (RX_CHF_COEF_5)	32	RW	0000_0000h
1B8h	Receive Channel Filter Coefficient 6 (RX_CHF_COEF_6)	32	RW	0000_0000h
1BCh	Receive Channel Filter Coefficient 7 (RX_CHF_COEF_7)	32	RW	0000_0000h
1C0h	Receive Channel Filter Coefficient 8 (RX_CHF_COEF_8)	32	RW	0000_0000h
1C4h	Receive Channel Filter Coefficient 9 (RX_CHF_COEF_9)	32	RW	0000_0000h
1C8h	Receive Channel Filter Coefficient 10 (RX_CHF_COEF_10)	32	RW	0000_0000h
1CCh	Receive Channel Filter Coefficient 11 (RX_CHF_COEF_11)	32	RW	0000_0000h
1D0h	AGC Manual AGC Index (AGC_MAN_AGC_IDX)	32	RW	0000_0000h
1D4h	DC Residual Control (DC_RESID_CTRL)	32	RW	0000_0000h
1D8h	DC Residual Estimate (DC_RESID_EST)	32	RO	0000_0000h
1DCh	RX RC Calibration Control0 (RX_RCCAL_CTRL0)	32	RW	0000_0000h
1E0h	RX RC Calibration Control1 (RX_RCCAL_CTRL1)	32	RW	0000_0000h
1E4h	RX RC Calibration Status (RX_RCCAL_STAT)	32	RO	0210_4210h
1E8h	Aux PLL Frequency Calibration Control (AUXPLL_FCAL_CTRL)	32	RW	0040_0000h
1ECh	Aux PLL Frequency Calibration Count 6 (AUXPLL_FCAL_CNT6)	32	RO	0000_0000h
1F0h	Aux PLL Frequency Calibration Count 5 and 4 (AUXPLL_FCAL_CNT5_4)	32	RO	0000_0000h
1F4h	Aux PLL Frequency Calibration Count 3 and 2 (AUXPLL_FCAL_CNT3_2)	32	RO	0000_0000h
1F8h	Aux PLL Frequency Calibration Count 1 and 0 (AUXPLL_FCAL_CNT1_0)	32	RO	0000_0000h

A.2.3.2 RX Digital Control (RX_DIG_CTRL)

A.2.3.2.1 Offset

Register	Offset
RX_DIG_CTRL	0h

A.2.3.2.2 Diagram



A.2.3.2.3 Fields

Field	Function
31 RX_DEC_FILT_SAT_Q	Decimator output, saturation detected for Q channel This bit will be set if a saturation condition is detected in the decimator filter Q channel. This bit is cleared by tsm_rx_dcoc_init and rx_init. 0b - A saturation condition has not occurred. 1b - A saturation condition has occurred.
30 RX_DEC_FILT_SAT_I	Decimator output, saturation detected for I channel This bit will be set if a saturation condition is detected in the decimator filter I channel. This bit is cleared by tsm_rx_dcoc_init and rx_init. 0b - A saturation condition has not occurred. 1b - A saturation condition has occurred.
29 RX_RSSI_FILT_HAZARD	Decimator output for RSSI, hazard condition detected This bit will be set if a hazard condition is detected in either the I or Q decimator filter related to the wideband RSSI measurement. This does not indicate that a data corruption has occurred, only that the conditions associated with data corruption were detected. This bit is cleared by rx_init. 0b - A hazard condition has not been detected

Table continues on the next page...

Field	Function
	1b - A hazard condition has been detected
28 RX_DEC_FILT_HAZARD	Decimator output, hazard condition detected This bit will be set if a hazard condition is detected in either the I or Q decimator filter. This does not indicate that a data corruption has occurred, only that the conditions associated with data corruption were detected. This bit is cleared by rx_init. 0b - A hazard condition has not been detected 1b - A hazard condition has been detected
27-26 —	Reserved.
25 RX_DEC_FILT_HZD_CORR_DIS	Decimator filter hazard correction disable This bit should be set for normal operation.
24-20 RX_DEC_FILT_GAIN	Decimation Filter Fractional Gain Defines the fractional gain which is applied at the decimator output. The format is u0.5. The gain applied is 1 + RX_DEC_FILT_GAIN/32, so e.g. if RX_DEC_FILT_GAIN is 5'b10110=5'd22, the gain is 1+22/32=1.6875. The nominal gain through the ADC is 2048codes/1.7V; the combined gain through the ADC and decimator would therefore be 2048codes/1.7e3mV*1.6875=2.03codes/mV for the example RX_DEC_FILT_GAIN=5'b10110.
19 —	Reserved.
18 RX_DMA_DTEST_EN	RX DMA and DTEST enable This bit should be set to ensure that all of the rx_dig outputs related to DMA or DTEST are enabled. In mission mode this bit is intended to be cleared to reduce switching power.
17 RX_SRC_RATE	RX Sample Rate Converter Rate Selections 0b - SRC is configured for a First Order Hold rate of 8/13. 1b - SRC is configured for a Zero Order Hold rate of 12/13.
16 RX_SRC_EN	RX Sample Rate Converter Enable 0b - SRC is disabled. 1b - SRC is enabled.
15 RX_DC_RESID_EN	DC Residual Enable Enables DC Residual block 0b - DC Residual block is disabled. 1b - DC Residual block is enabled.
14 RX_IQ_SWAP	RX IQ Swap Enable swap of I/Q channels (does not affect ADC raw mode). 0b - IQ swap is disabled. 1b - IQ swap is enabled.
13 RX_DCOC_CAL_EN	DCOC Calibration Enable Enable DCOC warm-up calibration in receiver. 0b - DCOC calibration is disabled. 1b - DCOC calibration is enabled.
12 RX_DCOC_EN	DCOC Enable Enables DCO calculation and application of corrections. 0b - DCOC is disabled. 1b - DCOC is enabled.
11	AGC Global Enable

Table continues on the next page...

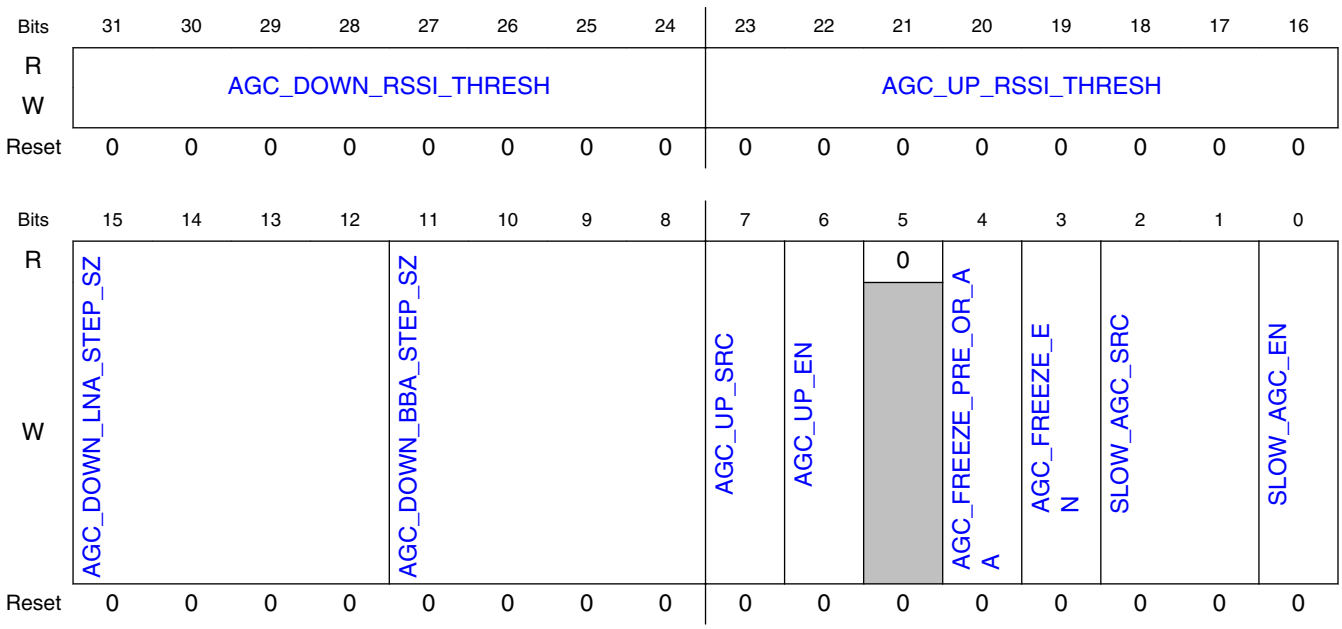
Field	Function
RX_AGC_EN	Does NOT affect user gains (user gain programming has priority). 0b - AGC is disabled. 1b - AGC is enabled.
10 RX_RSSI_EN	RSSI Measurement Enable 0b - RSSI measurement is disabled. 1b - RSSI measurement is enabled.
9 RX_NORM_EN	Normalizer Enable 0b - Normalizer is disabled. 1b - Normalizer is enabled.
8 RX_FSK_ZB_SEL	FSK / 802.15.4 demodulator select Select between FSK and 802.15.4 demodulator. This is used in the RSSI and AGC to select trigger source input signals. 0b - FSK demodulator. 1b - 802.15.4 demodulator.
7 —	Reserved.
6-4 RX_DEC_FILT_OSR	Decimation Filter Oversampling **NOTE:** All undocumented values are Reserved. 000b - OSR 4 001b - OSR 8 010b - OSR 16 011b - OSR 6 100b - OSR 32 101b - OSR 12 110b - OSR 24
3 RX_ADC_POL	Receive ADC Polarity Selects polarity of the ADC data 0b - ADC output of 1'b0 maps to -1, 1'b1 maps to +1 (default) 1b - ADC output of 1'b0 maps to +1, 1'b1 maps to -1
2 —	Reserved.
1 RX_CH_FILT_BYPASS	Receive Channel Filter Bypass Selects whether to disable and bypass channel filter. 0b - Channel filter is enabled. 1b - 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. 0b - Register ADC data on positive edge of clock 1b - Register ADC data on negative edge of clock

A.2.3.3 AGC Control 0 (AGC_CTRL_0)

A.2.3.3.1 Offset

Register	Offset
AGC_CTRL_0	4h

A.2.3.3.2 Diagram



A.2.3.3.3 Fields

Field	Function
31-24 AGC_DOWN_RSSI_THRESH	AGC DOWN RSSI Threshold ADC RSSI threshold to take downward step (AGC slow). If the ADC RSSI measurment is higher than this threshold, a downward gain step can be taken. The ADC RSSI is measured directly from the ADC input and does not have any adjustment applied for gain settings, etc. In other words, it is a measurement of how saturated the digital bus is, similar to a dBFS measurement.
23-16 AGC_UP_RSSI_THRESH	AGC UP RSSI Threshold ADC RSSI threshold to take upward step (AGC slow). If the ADC RSSI measurment is below this threshold, an upward gain step can be taken. The ADC RSSI is measured directly from the ADC input and does not have any adjustment applied for gain settings, etc. In other words, it is a measurement of how saturated the digital bus is, similar to a dBFS measurement.
15-12 AGC_DOWN_LNA_STEP_SZ	AGC_DOWN_LNA_STEP_SZ Number of table steps for downward step (LNA) in AGC fast.
11-8	AGC_DOWN_BBA_STEP_SZ Number of table steps for downward step (BBA) in AGC fast.

Table continues on the next page...

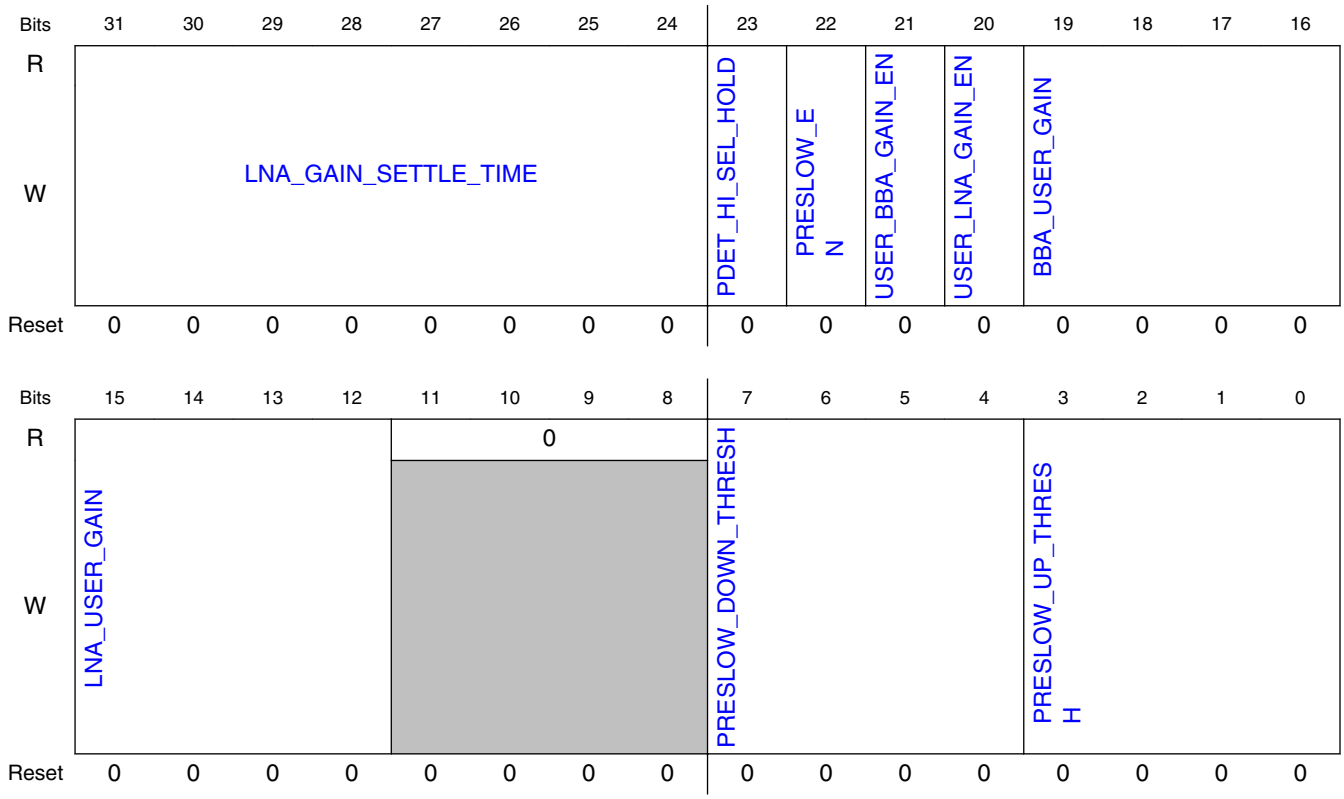
Field	Function
AGC_DOWN_BBA_STEP_SZ	
7 AGC_UP_SRC	AGC Up Source Criterion to use for upward AGC steps in SLOW state. For pdet_lo, a timing window is observed for no assertions of the LO peak detectors and then an upward step is made. If RSSI is selected, the current RSSI measurement is compared with the UP threshold to determine if an upward step is due. 0b - PDET LO 1b - RSSI
6 AGC_UP_EN	AGC Up Enable Allow AGC to take upward steps in the gain table in slow mode.
5 —	Reserved.
4 AGC_FREEZE_PRE_OR_AA	AGC Freeze Source Selection Select trigger source for entering freeze AGC (HOLD state). 0b - Access Address match (for active protocol) 1b - Preamble Detect (for active protocol)
3 AGC_FREEZE_EN	AGC Freeze Enable Allow AGC to freeze (ie enter HOLD state). 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. For address match and preamble detect, the trigger is generated by the PHY. If the Fast AGC expire timer is selected, when the timer expires, the AGC state machine will transition into SLOW. 00b - Access Address match (for active protocol) 01b - Preamble Detect (for active protocol) 10b - Fast AGC expire timer 11b - Reserved
0 SLOW_AGC_EN	Slow AGC Enable Allow AGC to enter into slow mode.

A.2.3.4 AGC Control 1 (AGC_CTRL_1)

A.2.3.4.1 Offset

Register	Offset
AGC_CTRL_1	8h

A.2.3.4.2 Diagram



A.2.3.4.3 Fields

Field	Function
31-24 LNA_GAIN_SETTLE_TIME	LNA_GAIN_SETTLE_TIME At LNA gain change, number of clocks to assert TZA peak detector reset (for automatic control). Should be programmed greater than zero.
23 PDET_HI_SEL_HOLD	AGC HOLD hysteresis When enabled, and the AGC state machine is in HOLD, the BBA clip detector threshold is raised by 1 count. 0b - Disabled. 1b - Enabled.
22 PRESLOW_EN	Pre-slow Enable Enable the pre-slow state where signal headroom is checked before entering SLOW. 0b - Pre-slow is disabled. 1b - Pre-slow is enabled.
21 USER_BBA_GAIN_EN	User BBA Gain Enable Enable user defined BBA gain (no AGC).
20 USER_LNA_GAIN_EN	User LNA Gain Enable Enable user defined LNA gain (no AGC).

Table continues on the next page...

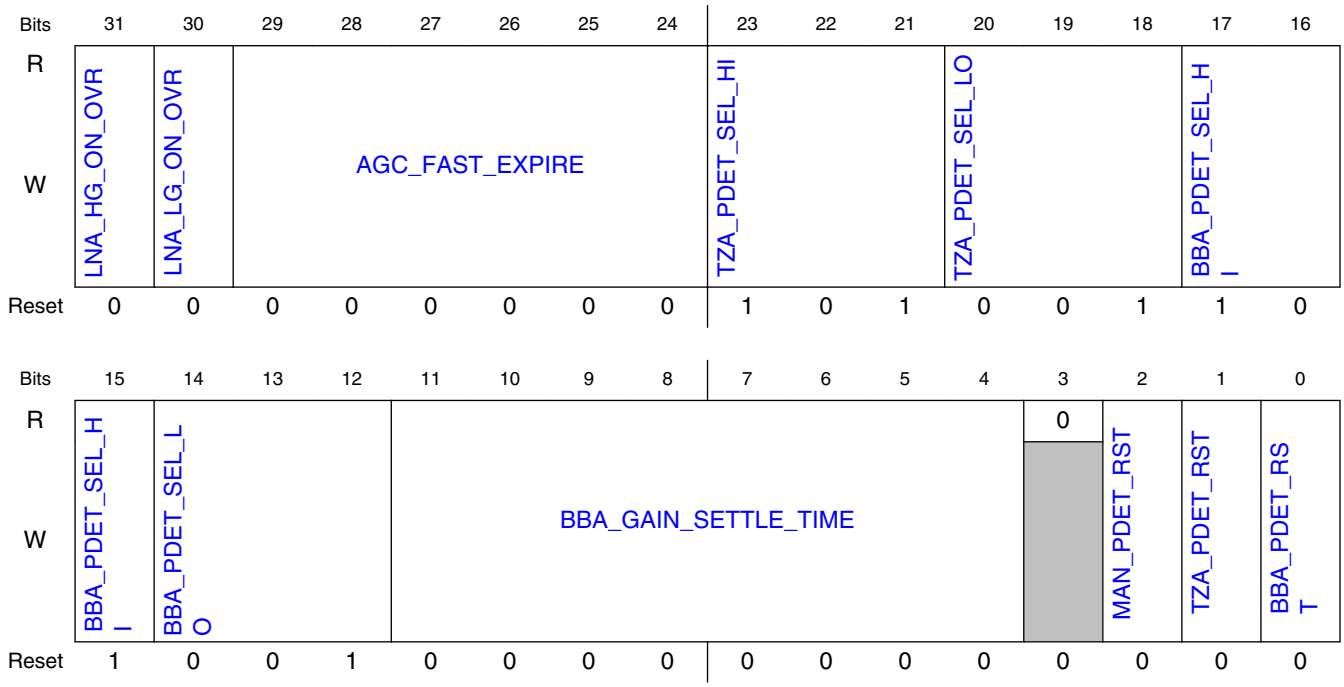
Field	Function
19-16 BBA_USER_GAIN	BBA_USER_GAIN User defined BBA gain index if user_bba_gain_en =1
15-12 LNA_USER_GAIN	LNA_USER_GAIN User defined LNA gain index if user_lna_gain_en =1.
11-8 —	Reserved.
7-4 PRESLOW_DOWN_THRESH	PRESLOW_DOWN_THRESH ADC RSSI threshold to take downward step in (AGC PRESLOW state). The actual threshold is computed as the programmed value minus 10, so the threshold assumes values in the range [-10,5]. If the ADC RSSI measurment is above this threshold, a downward gain step of 1 is taken. The ADC RSSI is measured directly from the ADC input and does not have any adjustment applied for gain settings, etc. In other words, it is a measurement of how saturated the digital bus is, similar to a dBFS measurement.
3-0 PRESLOW_UP_THRESH	PRESLOW_UP_THRESH ADC RSSI threshold to take upward step in (AGC PRESLOW state). The actual threshold is computed as the programmed value minus 15, so the threshold assumes values in the range [-15,0]. If the ADC RSSI measurment is below this threshold, an upward gain step of 1 is taken. The ADC RSSI is measured directly from the ADC input and does not have any adjustment applied for gain settings, etc. In other words, it is a measurement of how saturated the digital bus is, similar to a dBFS measurement.

A.2.3.5 AGC Control 2 (AGC_CTRL_2)

A.2.3.5.1 Offset

Register	Offset
AGC_CTRL_2	Ch

A.2.3.5.2 Diagram



A.2.3.5.3 Fields

Field	Function
31 LNA_HG_ON_OVR	LNA_HG_ON override If set, the lna high gain "on" signal is always asserted. Otherwise, this signal is asserted automatically based on the LNA_GAIN code.
30 LNA_LG_ON_OVR	LNA_LG_ON override If set, the lna low gain "on" signal is always asserted. Otherwise, this signal is asserted automatically based on the LNA_GAIN code.
29-24 AGC_FAST_EXPIRE	AGC Fast Expire Expire time (uS) for fast AGC (1-63uS).
23-21 TZA_PDET_SEL_HI	TZA PDET Threshold High TZA peak detect HI threshold. 000b - 0.600V 001b - 0.645V 010b - 0.705V 011b - 0.750V 100b - 0.795V 101b - 0.855V 110b - 0.900V 111b - 0.945V
20-18	TZA PDET Threshold Low TZA peak detect LO threshold.

Table continues on the next page...

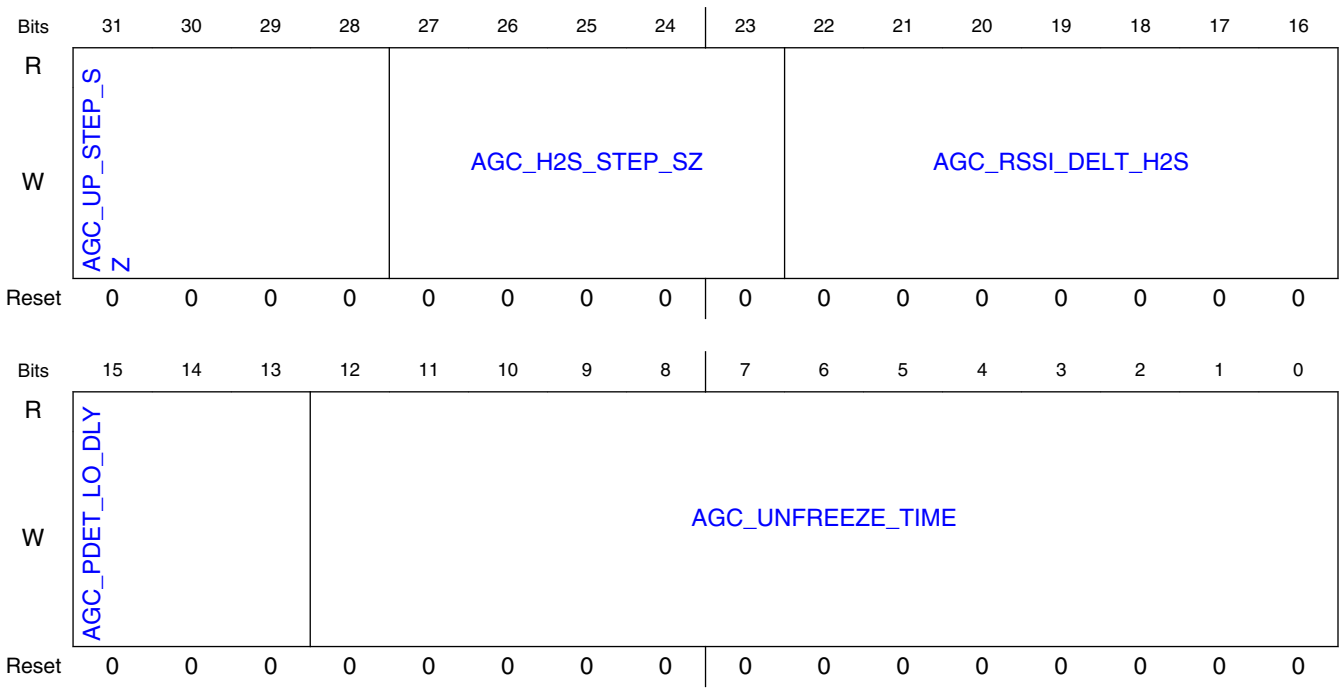
Field	Function
TZA_PDET_SEL_LO	000b - 0.600V 001b - 0.615V 010b - 0.630V 011b - 0.645V 100b - 0.660V 101b - 0.675V 110b - 0.690V 111b - 0.705V
17-15 BBA_PDET_SEL_HI	BBA PDET Threshold High BBA peak detect HI threshold. 000b - 0.600V 001b - 0.795V 010b - 0.900V 011b - 0.945V 100b - 1.005V 101b - 1.050V 110b - 1.095V 111b - 1.155V
14-12 BBA_PDET_SEL_LO	BBA PDET Threshold Low BBA peak detect LO threshold. 000b - 0.600V 001b - 0.615V 010b - 0.630V 011b - 0.645V 100b - 0.660V 101b - 0.675V 110b - 0.690V 111b - 0.705V
11-4 BBA_GAIN_SETTLE_TIME	BBA Gain Settle Time Number of clocks to assert BBA peak detector reset (for automatic control). Should be programmed greater than zero.
3 —	Reserved.
2 MAN_PDET_RST	MAN PDET Reset 0b - The peak detector reset signals are controlled automatically by the AGC. 1b - The BBA_PDET_RST and TZA_PDET_RST are used to manually control the peak detector reset signals.
1 TZA_PDET_RST	TZA PDET Reset TZA peak detector reset, manual control.
0 BBA_PDET_RST	BBA PDET Reset BBA peak detector reset, manual control.

A.2.3.6 AGC Control 3 (AGC_CTRL_3)

A.2.3.6.1 Offset

Register	Offset
AGC_CTRL_3	10h

A.2.3.6.2 Diagram



A.2.3.6.3 Fields

Field	Function
31-28 AGC_UP_STEP_SZ	AGC Up Step Size Number of AGC gain table steps for upward step
27-23 AGC_H2S_STEP_SZ	AGC_H2S_STEP_SZ AGC gain table 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. This delta is observed from a previous RSSI measurement to a current measurement.
15-13 AGC_PDET_LO_DLY	AGC Peak Detect Low Delay Time (uS) to wait for pdet low to assert (1-7uS).
12-0	AGC Unfreeze Time

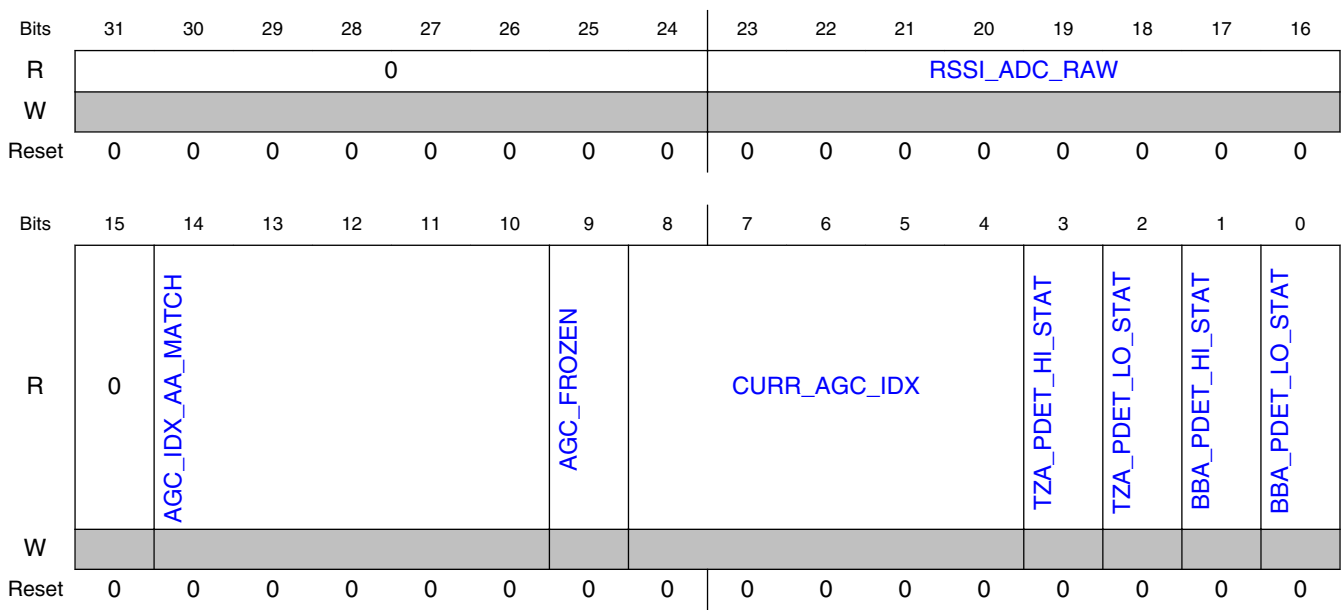
Field	Function
AGC_UNFREEZE_TIME	Time (uS) for AGC to unfreeze (1-8191uS) from HOLD and re-enter SLOW state.

A.2.3.7 AGC Status (AGC_STAT)

A.2.3.7.1 Offset

Register	Offset
AGC_STAT	14h

A.2.3.7.2 Diagram



A.2.3.7.3 Fields

Field	Function
31-24 —	Reserved.
23-16 RSSI_ADC_RAW	ADC RAW RSSI Reading Reading of ADC rssi (before adjustments)

Table continues on the next page...

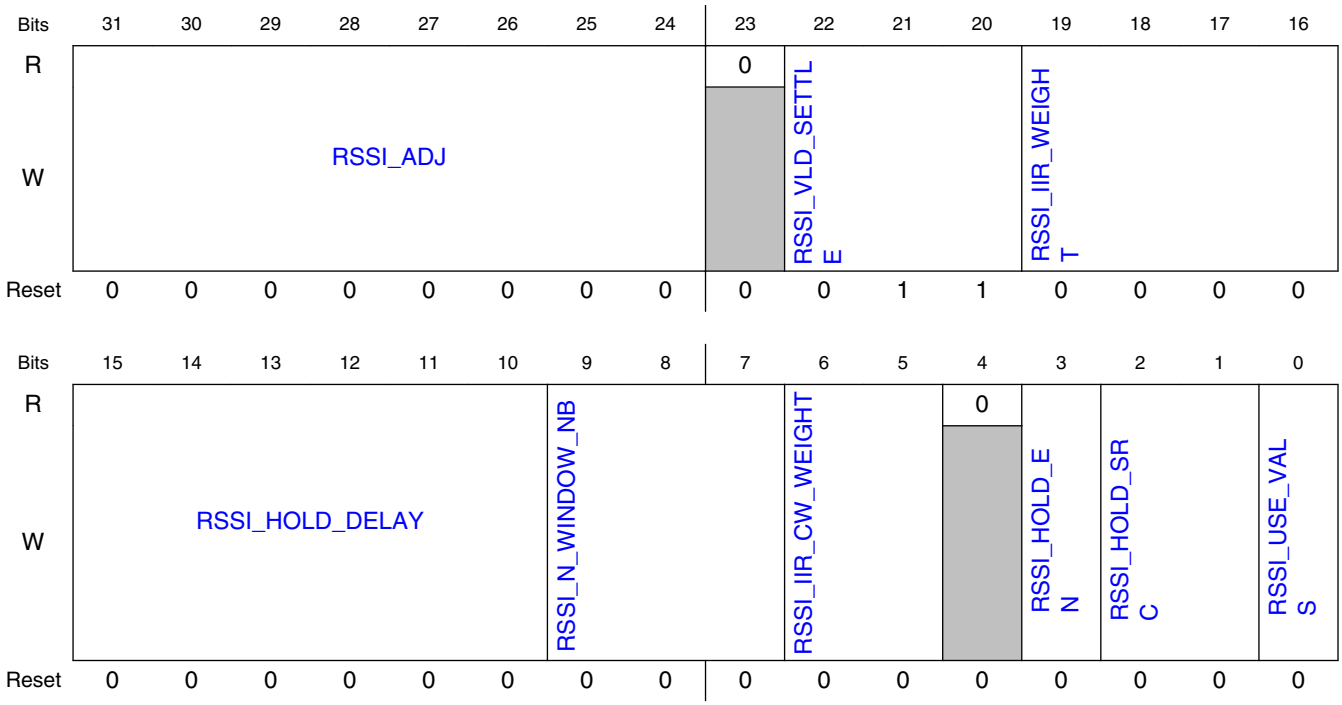
Field	Function
15 —	Reserved.
14-10 AGC_IDX_AA_MATCH	AGC Gain Index at AA Match AGC Gain Index captured at AA Match. This value remains static so it can be read by software after the end of the RX burst. The value is cleared to 0 at TSM rx_init and radio reset conditions.
9 AGC_FROZEN	AGC Frozen Status Status of AGC freeze. 0b - AGC is not frozen. 1b - 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 BBA_PDET_HI_STAT	BBA Peak Detector High Status Status of BBA peak detector HI flag (1=set)
0 BBA_PDET_LO_STAT	BBA Peak Detector Low Status Status of BBA peak detector LO flag (1=set)

A.2.3.8 RSSI Control 0 (RSSI_CTRL_0)

A.2.3.8.1 Offset

Register	Offset
RSSI_CTRL_0	18h

A.2.3.8.2 Diagram



A.2.3.8.3 Fields

Field	Function
31-24 RSSI_ADJ	RSSI Adjustment RSSI calculation adjustment (8-bit signed 1/4 dB).
23 —	Reserved.
22-20 RSSI_VLD_SETTLE	RSSI Valid Settle Sets number of us (times 8) that RSSI will be considered invalid after certain events.
19-16 RSSI_IIR_WEIGHT	RSSI IIR Weighting IIR filter weight for RSSI filtering. **NOTE:** All undocumented values are Reserved. 0000b - Bypass 0001b - 1/2 0010b - 1/4 0011b - 1/8 0100b - 1/16 0101b - 1/32
15-10 RSSI_HOLD_DELAY	RSSI Hold Delay Sets number of us (times 8) that RSSI will run after a hold event before the value is frozen.
9-7	RSSI N Window Average Narrowband

Table continues on the next page...

Field	Function
RSSI_N_WINDOW_NB	Selects Averaging window length for RSSI in Narrowband mode. 000b - No averaging 001b - Averaging window length is 2 samples 010b - Averaging window length is 4 samples 011b - Averaging window length is 8 samples 100b - Averaging window length is 16 samples 101b - Averaging window length is 32 samples
6-5 RSSI_IIR_CW_WEIGHT	RSSI IIR CW Weighting IIR filter weight for RSSI filtering of a CW input. 00b - Bypass 01b - 1/8 10b - 1/16 11b - 1/32
4 —	Reserved.
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	RSSI Hold Source Selection Select trigger source for freezing RSSI measurement. 00b - Access Address match 01b - Preamble Detect 10b - Reserved 11b - 802.15.4 LQI done (1=freeze, 0=run AGC)
0 RSSI_USE_VALS	RSSI Values Selection Enable use of LNA and BBA gain values programmed in registers for calculation.

A.2.3.9 RSSI Control 1 (RSSI_CTRL_1)

A.2.3.9.1 Offset

Register	Offset
RSSI_CTRL_1	1Ch

A.2.3.9.2 Diagram

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

A.2.3.9.3 Fields

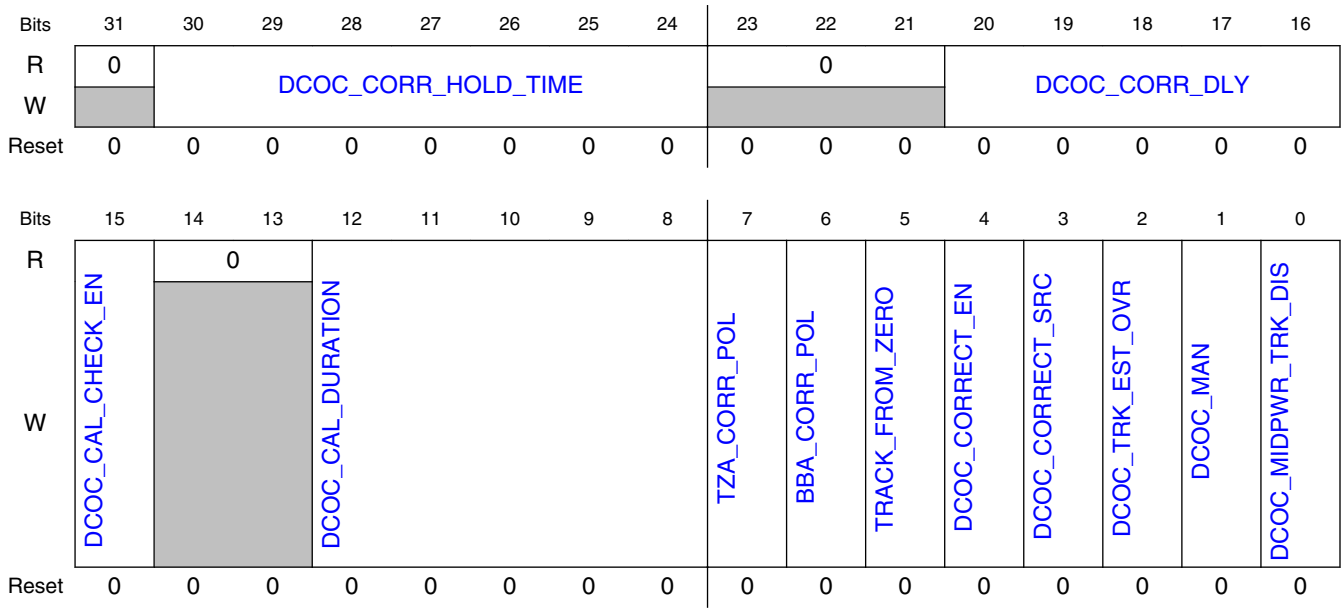
Field	Function
31-24 RSSI_OUT	RSSI Reading RSSI output (8-bit signed).
23-0 —	Reserved.

A.2.3.10 DCOC Control 0 (DCOC_CTRL_0)

A.2.3.10.1 Offset

Register	Offset
DCOC_CTRL_0	24h

A.2.3.10.2 Diagram



A.2.3.10.3 Fields

Field	Function
31 —	Reserved.
30-24 DCOC_CORR_HOLD_TIME	DCOC Correction Hold Time Delay from last gain change to freezing the DC correction. 0000000b - Reserved 0000001-1111110b - For a 32MHz reference clock, this is the delay in microseconds; for other reference clock frequencies, the delay is scaled accordingly. 1111111b - The DC correction is not frozen.
23-21 —	Reserved.
20-16 DCOC_CORR_DLY	DCOC Correction Delay Wait time between corrections. 00000b - Reserved 00001-11111b - For a 32MHz reference clock, this is the wait time in microseconds; for other reference clock frequencies, the delay is scaled accordingly.
15 DCOC_CAL_CHECK_EN	DCOC Calibration Check Enable 0b - Calibration checking disabled. The DCOC_OFFSET_n registers are always updated during calibration. 1b - Calibration checking enabled. The DCOC_OFFSET_n registers are updated conditionally depending on the outcome of the pass/fail threshold checks performed on the alpha-hat and beta-hat estimates during calibration.
14-13 —	Reserved.

Table continues on the next page...

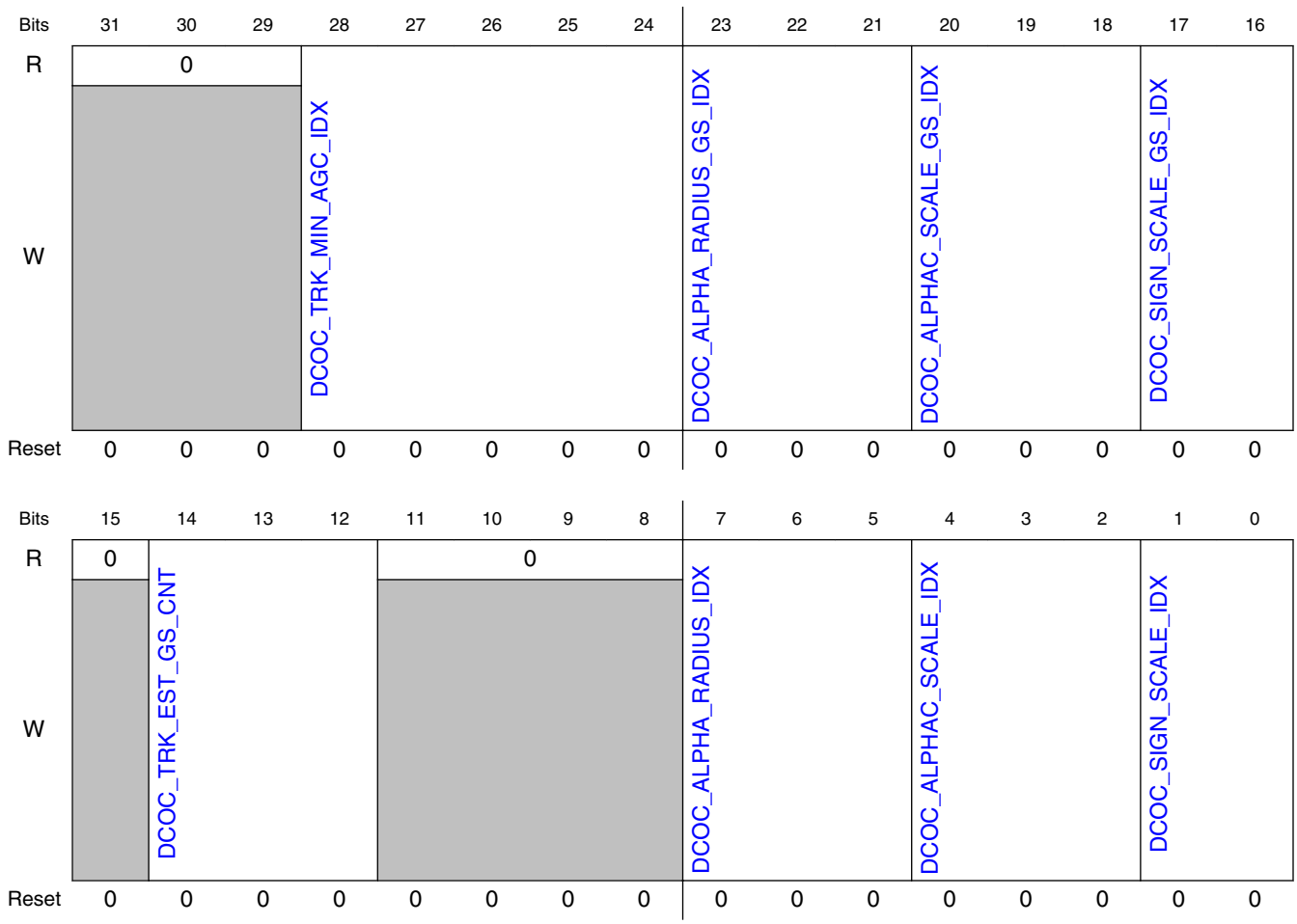
Field	Function
12-8 DCOC_CAL_DURATION	DCOC Calibration Duration 00000b - Reserved 00001-11111b - For a 32MHz reference clock, this is the calibration duration in microseconds; for other reference clock frequencies, the delay is scaled accordingly.
7 TZA_CORR_POL	TZA Correction Polarity Selects polarity of TZA corrections. 0b - Normal polarity. 1b - Negative polarity. This should be set if the ADC output is inverted, or if the TZA DACs were implemented with negative polarity.
6 BBA_CORR_POL	BBA Correction Polarity Selects polarity of BBA corrections. 0b - Normal polarity. 1b - Negative polarity. This should be set if the ADC output is inverted, or if the BBA DACs were implemented with negative polarity.
5 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. 0b - Track from current I/Q sample. 1b - Track from zero.
4 DCOC_CORRECT_EN	DCOC Correction Enable 0b - Correction disabled. The DCOC will not correct the DC offset. 1b - Correction enabled. The DCOC will use the TZA and BBA DACs, and apply digital corrections (if DCOC_CORRECT_SRC=1) to correct the DC offset.
3 DCOC_CORRECT_SRC	DCOC Corrector Source If not set, the corrector uses only the DCOC calibration table to apply corrections to the DCOC DACs. 0b - If correction is enabled, the DCOC will use only the DCOC calibration table to correct the DC offset. 1b - If correction is enabled, the DCOC will use the DCOC calibration table and then the tracking estimator to correct the DC offset.
2 DCOC_TRK_EST_OVR	Override for the DCOC tracking estimator 0b - The tracking estimator is enabled only as needed by the corrector 1b - The tracking estimator remains enabled whenever the DCOC is active
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 DCOC_DAC_INIT and DCOC_DIG_MAN, respectively.
0 DCOC_MIDPWR_TRK_DIS	DCOC Mid Power Tracking Disable Disables tracking correction at mid power levels, as indicated by the TZA and BBA lo peak detectors. This is implemented by reseting the counters associated with DCOC_CORR_DLY and DCOC_CORR_HOLD_TIME, so if the lo peak detectors do not continue to assert, tracking corrections would resume operation. The tracking estimator is not disabled and the counter associated with DCOC_TRK_EST_GS_CNT is not reset when this condition occurs. 0b - Tracking corrections are enabled as determined by DCOC_CORRECT_SRC and DCOC_TRK_MIN_AGC_IDX. 1b - Tracking corrections are disabled when either the TZA or BBA lo peak detector asserts.

A.2.3.11 DCOC Control 1 (DCOC_CTRL_1)

A.2.3.11.1 Offset

Register	Offset
DCOC_CTRL_1	28h

A.2.3.11.2 Diagram



A.2.3.11.3 Fields

Field	Function
31-29 —	Reserved.
28-24	DCOC Tracking Minimum AGC Table Index

Table continues on the next page...

Field	Function
DCOC_TRK_MIN_AGC_IDX	Specifies the minimum AGC table index value at which tracking is enabled. E.g., if this is 5'd0, then tracking is enabled for all AGC gain table indexes (assuming DCOC_CORRECT_SRC is set) if this is 5'd20, then tracking is enabled for AGC gain table indexes 20-26 (assuming DCOC_CORRECT_SRC is set).
23-21 DCOC_ALPHA_RADIUS_GS_IDX	Alpha-R Scaling for Gearshift DCOC Alpha-R Scaling for Gearshift. Radius stepsize used in the DCOC tracking estimator after the gearshift counter has expired. Used when DCOC_CORRECT_SRC=1. 000b - 1 001b - 1/2 010b - 1/4 011b - 1/8 100b - 1/16 101b - 1/32 110b - 1/64 111b - Reserved
20-18 DCOC_ALPHAC_SCALE_GS_IDX	DCOC Alpha-C Scaling for Gearshift DCOC Alpha-C Scaling for Gearshift. I/Q center stepsize used in the DCOC tracking estimator after the gearshift counter has expired. Used when DCOC_CORRECT_SRC=1. 000b - 1/2 001b - 1/4 010b - 1/8 011b - 1/16 100b - 1/32 101b - 1/64 110b - Reserved 111b - Reserved
17-16 DCOC_SIGN_SCALE_GS_IDX	DCOC Sign Scaling for Gearshift DCOC Sign Scaling for Gearshift. Sign()-based scaling factor used in the DCOC tracking estimator after the gearshift counter has expired. Used when DCOC_CORRECT_SRC=1. 00b - 1/8 01b - 1/16 10b - 1/32 11b - 1/64
15 —	Reserved.
14-12 DCOC_TRK_EST_GS_CNT	DCOC Tracking Estimator Gearshift Count Specifies the number of corrections (periods of DCOC_CORR_DLY) before the tracking estimator switches from using the set of parameters {DCOC_ALPHA_RADIUS_IDX, DCOC_ALPHAC_SCALE_IDX, DCOC_SIGN_SCALE_IDX} to the set of gearshift parameters {DCOC_ALPHA_RADIUS_GS_IDX, DCOC_ALPHAC_SCALE_GS_IDX, DCOC_SIGN_SCALE_GS_IDX}. If the value is 0, the set of gearshift parameters are not used.
11-8 —	Reserved.
7-5 DCOC_ALPHA_RADIUS_IDX	Alpha-R Scaling DCOC Alpha-R Scaling. Radius stepsize used in the DCOC tracking estimator. Used when DCOC_CORRECT_SRC=1. 000b - 1 001b - 1/2 010b - 1/4 011b - 1/8 100b - 1/16

Table continues on the next page...

Field	Function
	101b - 1/32 110b - 1/64 111b - Reserved
4-2 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_CORRECT_SRC=1. 000b - 1/2 001b - 1/4 010b - 1/8 011b - 1/16 100b - 1/32 101b - 1/64 110b - Reserved 111b - Reserved
1-0 DCOC_SIGN_SCALE_IDX	DCOC Sign Scaling DCOC Sign Scaling. Sign()-based scaling factor used in the DCOC tracking estimator. Used when DCOC_CORRECT_SRC=1. 00b - 1/8 01b - 1/16 10b - 1/32 11b - 1/64

A.2.3.12 DCOC DAC Initialization (DCOC_DAC_INIT)

A.2.3.12.1 Offset

Register	Offset
DCOC_DAC_INIT	2Ch

A.2.3.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TZA_DCOC_INIT_Q								TZA_DCOC_INIT_I							
Reset	1	0	0	0	0	0	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	0		BBA_DCOC_INIT_Q						0		BBA_DCOC_INIT_I					
W																
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

A.2.3.12.3 Fields

Field	Function
31-24 TZA_DCOC_INIT_Q	DCOC TZA Init Q Value used for DCOC TZA Q channel DAC during calibration and for manual override.
23-16 TZA_DCOC_INIT_I	DCOC TZA Init I Value used for DCOC TZA I channel DAC during calibration and for manual override.
15-14 —	Reserved.
13-8 BBA_DCOC_INIT_Q	DCOC BBA Init Q Value used for DCOC BBA Q channel DAC during calibration and for manual override.
7-6 —	Reserved.
5-0 BBA_DCOC_INIT_I	DCOC BBA Init I Value used for DCOC BBA I channel DAC during calibration and for manual override.

A.2.3.13 DCOC Digital Correction Manual Override (DCOC_DIG_MAN)

A.2.3.13.1 Offset

Register	Offset
DCOC_DIG_MAN	30h

A.2.3.13.2 Diagram

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

A.2.3.13.3 Fields

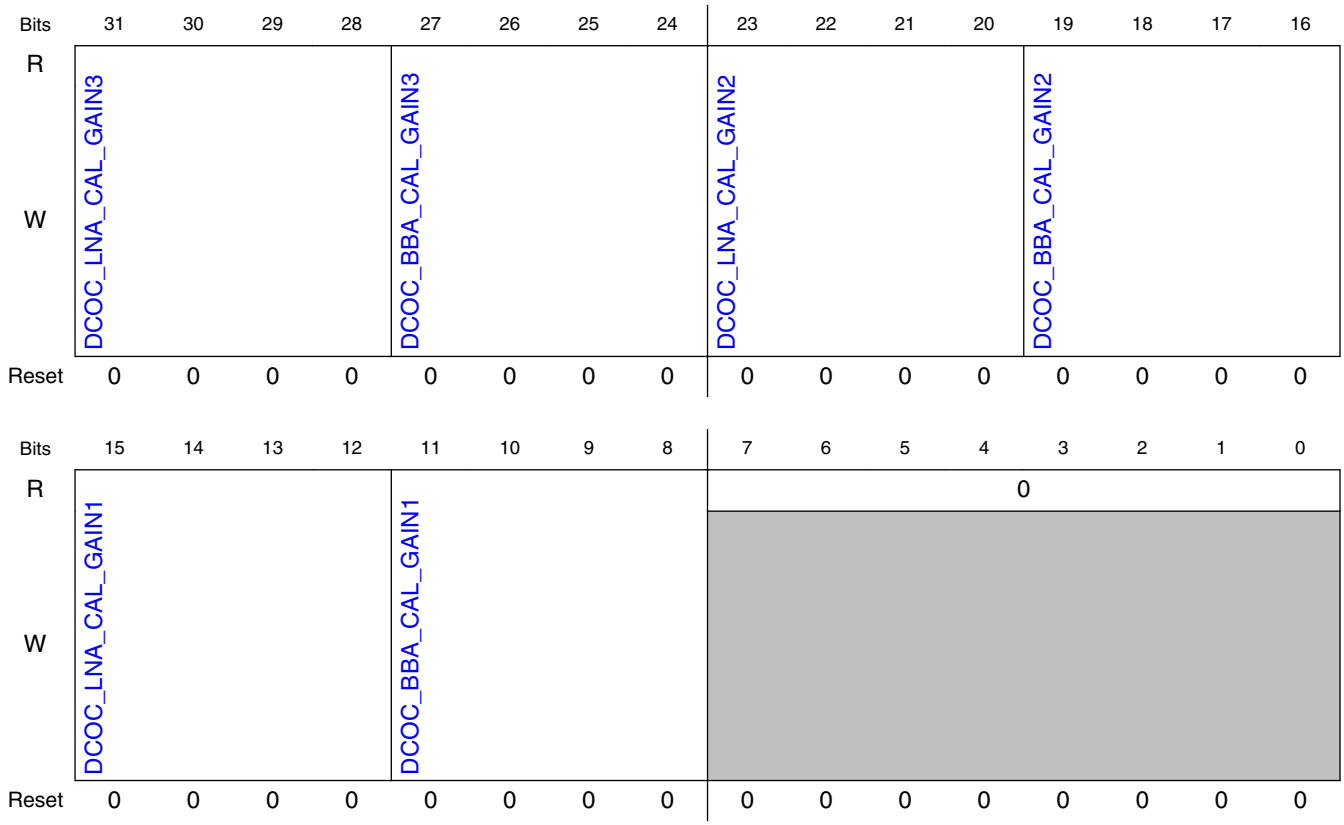
Field	Function
31-28 —	Reserved.
27-16 DIG_DCOC_INIT_Q	DCOC DIG Init Q Manual override for DCOC DIG Q channel correction. Value to be subtracted from downsampled Q channel. Used when DCOC_MAN=1.
15-12 —	Reserved.
11-0 DIG_DCOC_INIT_I	DCOC DIG Init I Manual override for DCOC DIG I channel correction. Value to be subtracted from downsampled I channel. Used when DCOC_MAN=1.

A.2.3.14 DCOC Calibration Gain (DCOC_CAL_GAIN)

A.2.3.14.1 Offset

Register	Offset
DCOC_CAL_GAIN	34h

A.2.3.14.2 Diagram



A.2.3.14.3 Fields

Field	Function
31-28 DCOC_LNA_CAL_GAIN3	DCOC LNA Calibration Gain 3 The LNA gain index used for the 3rd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
27-24 DCOC_BBA_CAL_GAIN3	DCOC BBA Calibration Gain 3 The BBA gain index used for the 3rd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
23-20 DCOC_LNA_CAL_GAIN2	DCOC LNA Calibration Gain 2 The LNA gain index used for the 2nd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
19-16 DCOC_BBA_CAL_GAIN2	DCOC BBA Calibration Gain 2 The BBA gain index used for the 2nd DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
15-12 DCOC_LNA_CAL_GAIN1	DCOC LNA Calibration Gain 1 The LNA gain index used for the 1st DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
11-8	DCOC BBA Calibration Gain 1

Table continues on the next page...

Field	Function
DCOC_BBA_CAL_GAIN1	The BBA gain index used for the 1st DCOC calibration point. Used when RX_DCOC_CAL_EN=1.
7-0 —	Reserved.

A.2.3.15 DCOC Status (DCOC_STAT)

A.2.3.15.1 Offset

Register	Offset
DCOC_STAT	38h

A.2.3.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TZA_DCOC_Q								TZA_DCOC_I							
W																
Reset	1	0	0	0	0	0	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	DCOC_CAL_RESULT		BBA_DCOC_Q						DCOC_CAL_GTWSR	0	BBA_DCOC_I					
W																
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

A.2.3.15.3 Fields

Field	Function
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.

Table continues on the next page...

Field	Function
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 DCOC_CAL_RESULT	DCOC_CAL_RESULT Indicates the calibration result. Only applicable when DCOC_CTRL_0[DCOC_CAL_CHECK_EN] is set. 00b - Calibration checks failed. DCOC_OFFSET_n tables not updated. 01b - Calibration checks neither passed nor failed, DCOC_OFFSET_n tables not updated. 10b - Calibration checks neither passed nor failed, DCOC_OFFSET_n tables updated since no previous Pass condition has occurred since the last radio reset. 11b - Calibration checks passed. DCOC_OFFSET_n tables updated
13-8 BBA_DCOC_Q	DCOC BBA DAC Q Current BBA DAC setting for Q channel. Note that the BBA 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 DCOC_CAL_GTWSR	DCOC calibration Good Table Written Since Reset Indicates whether a Passing calibration check has occurred since the last reset. Only applicable when DCOC_CTRL_0[DCOC_CAL_CHECK_EN] is set. 0b - A Passing calibration result has not occurred since the last radio reset. 1b - A Passing calibration result has occurred since the last radio reset.
6 —	Reserved.
5-0 BBA_DCOC_I	DCOC BBA DAC I Current BBA DAC setting for I channel. Note that the BBA 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.

A.2.3.16 DCOC DC Estimate (DCOC_DC_EST)

A.2.3.16.1 Offset

Register	Offset
DCOC_DC_EST	3Ch

A.2.3.16.2 Diagram

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

A.2.3.16.3 Fields

Field	Function
31-28 —	Reserved.
27-16 DC_EST_Q	DCOC DC Estimate Q Reflects the current DCOC DC tracking estimate for Q channel. Format s11. Used when DCOC_CORRECT_SRC=1. This is provided for debug and characterization purposes only.
15-12 —	Reserved.
11-0 DC_EST_I	DCOC DC Estimate I Reflects the current DCOC DC tracking estimate for I channel. Format s11. Used when DCOC_CORRECT_SRC=1. This is provided for debug and characterization purposes only.

A.2.3.17 DCOC Calibration Reciprocals (DCOC_CAL_RCP)

A.2.3.17.1 Offset

Register	Offset
DCOC_CAL_RCP	40h

A.2.3.17.2 Diagram

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

A.2.3.17.3 Fields

Field	Function
31-27 —	Reserved.
26-16 ALPHA_CALC_RECIP	Alpha Calculation Reciprocal DCOC Alpha calculation reciprocal (format: u.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 $round([1.0/((G_{L\_HI} - G_{L\_LO})*G_{B\_LO})]*2^{11})$.
15-11 —	Reserved.
10-0 DCOC_TMP_CALC_RECIP	DCOC Calculation Reciprocal DCOC_tmp calculation reciprocal (format: u1.10). 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 $round([1.0/(G_{B\_HI} - G_{B\_LO})]*2^{10})$.

A.2.3.18 IQMC Control (IQMC_CTRL)

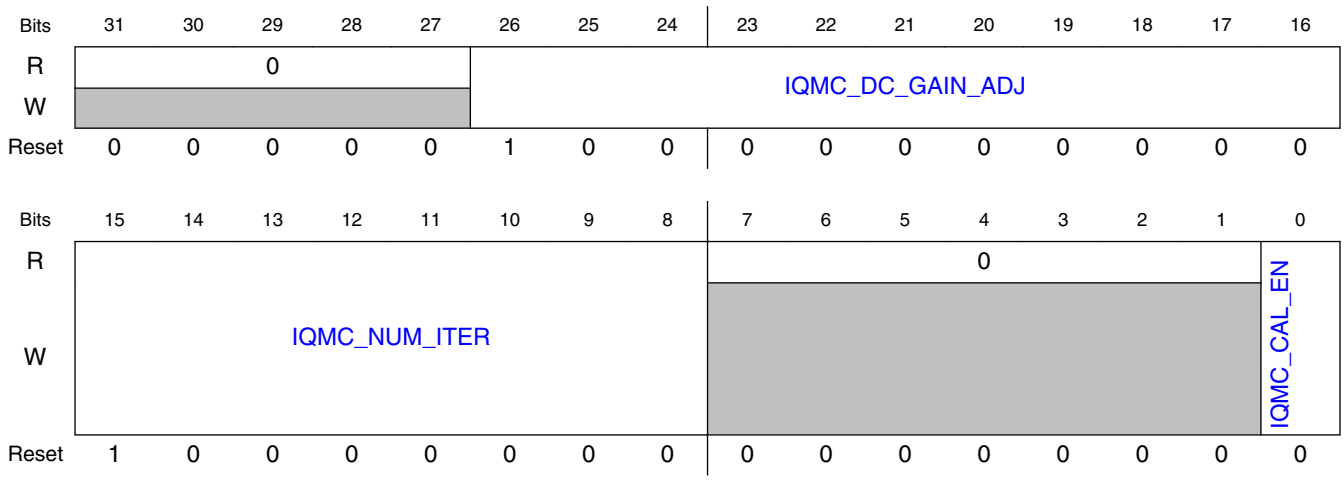
A.2.3.18.1 Offset

Register	Offset
IQMC_CTRL	48h

A.2.3.18.2 Function

IQMC Control register. This register can only be accessed when the radio oscillator clock is active.

A.2.3.18.3 Diagram



A.2.3.18.4 Fields

Field	Function
31-27 —	Reserved.
26-16 IQMC_DC_GAIN_ADJ	IQ Mismatch Correction DC Gain Coeff I/Q mismatch correction DC gain coefficient. This gain value is used only during DCOC calibration. It is not calculated during the IQMC calibration sequence, so it must be written by software. The format is u1.10; the reset value of 0x400 corresponds to a DC gain coefficient of 1.0.
15-8 IQMC_NUM_ITER	IQ Mismatch Cal Num Iter Number of iterations for IQ Mismatch Calibration.
7-1 —	Reserved.
0 IQMC_CAL_EN	IQ Mismatch Cal Enable Enables IQ mismatch calibration. This bit is self-clearing; it will clear automatically at the end of the calibration sequence.

A.2.3.19 IQMC Calibration (IQMC_CAL)

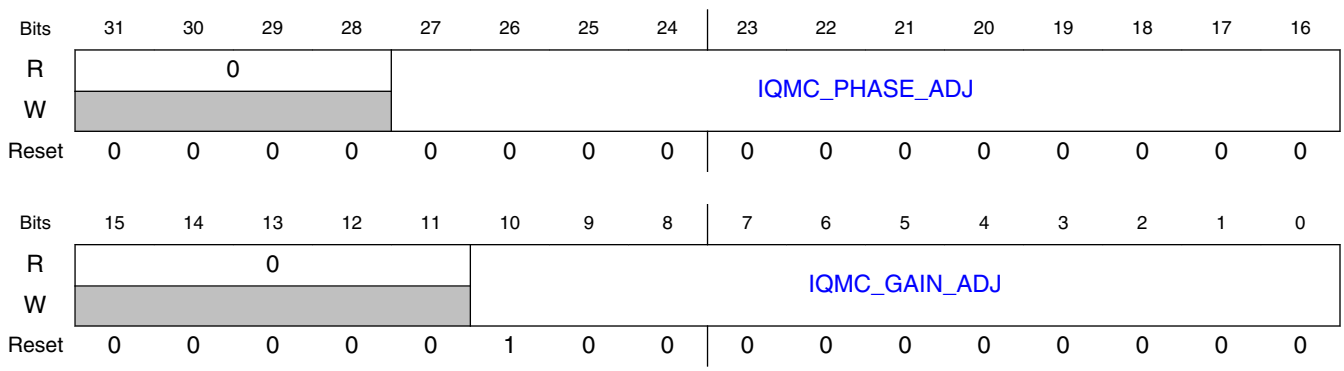
A.2.3.19.1 Offset

Register	Offset
IQMC_CAL	4Ch

A.2.3.19.2 Function

IQMC Calibration register. This register can only be accessed when the radio oscillator clock is active.

A.2.3.19.3 Diagram



A.2.3.19.4 Fields

Field	Function
31-28 —	Reserved.
27-16 IQMC_PHASE_ADJ	IQ Mismatch Correction Phase Coeff I/Q mismatch correction phase coefficient. This value is updated automatically at the end of the IQMC calibration sequence. It can also be written by software. The format is signed, with the maximum positive value 0x7ff corresponding to a phase coefficient of 0.25, and the maximum negative value 0x800 corresponding to -0.25. The reset value of 0x000 corresponds to a phase coefficient of 0.
15-11 —	Reserved.
10-0 IQMC_GAIN_ADJ	IQ Mismatch Correction Gain Coeff I/Q mismatch correction gain coefficient. This value is updated automatically at the end of the IQMC calibration sequence. It can also be written by software. The format is u1.10. The reset value of 0x400 corresponds to a gain coefficient of 1.0.

A.2.3.20 LNA_GAIN Step Values 3..0 (LNA_GAIN_VAL_3_0)

A.2.3.20.1 Offset

Register	Offset
LNA_GAIN_VAL_3_0	50h

A.2.3.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_VAL_3								LNA_GAIN_VAL_2							
Reset	0	0	1	1	1	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	LNA_GAIN_VAL_1								LNA_GAIN_VAL_0							
Reset	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0	1

A.2.3.20.3 Fields

Field	Function
31-24 LNA_GAIN_VAL_3	LNA_GAIN step 3 Gain for LNA gain Step 3, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
23-16 LNA_GAIN_VAL_2	LNA_GAIN step 2 Gain for LNA gain Step 2, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
15-8 LNA_GAIN_VAL_1	LNA_GAIN step 1 Gain for LNA gain Step 1, used in RSSI calculation. Unsigned, 4*(measured_gain_dB+16).
7-0 LNA_GAIN_VAL_0	LNA_GAIN step 0 Gain for LNA gain Step 0, used in RSSI calculation. Unsigned, 4*(measured_gain_dB+16).

A.2.3.21 LNA_GAIN Step Values 7..4 (LNA_GAIN_VAL_7_4)

A.2.3.21.1 Offset

Register	Offset
LNA_GAIN_VAL_7_4	54h

A.2.3.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_VAL_7								LNA_GAIN_VAL_6							
Reset	1	0	0	0	1	0	1	1	0	1	1	1	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LNA_GAIN_VAL_5								LNA_GAIN_VAL_4							
Reset	0	1	0	1	1	1	0	1	0	1	0	0	1	1	1	1

A.2.3.21.3 Fields

Field	Function
31-24 LNA_GAIN_VAL_7	LNA_GAIN step 7 Gain for LNA gain Step 7, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
23-16 LNA_GAIN_VAL_6	LNA_GAIN step 6 Gain for LNA gain Step 6, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
15-8 LNA_GAIN_VAL_5	LNA_GAIN step 5 Gain for LNA gain Step 5, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
7-0 LNA_GAIN_VAL_4	LNA_GAIN step 4 Gain for LNA gain Step 4, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).

A.2.3.22 LNA_GAIN Step Values 8 (LNA_GAIN_VAL_8)

A.2.3.22.1 Offset

Register	Offset
LNA_GAIN_VAL_8	58h

A.2.3.22.2 Diagram

Bits	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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LNA_GAIN_VAL_9								LNA_GAIN_VAL_8							
W																
Reset	1	0	1	1	0	1	1	0	1	0	1	0	0	0	0	1

A.2.3.22.3 Fields

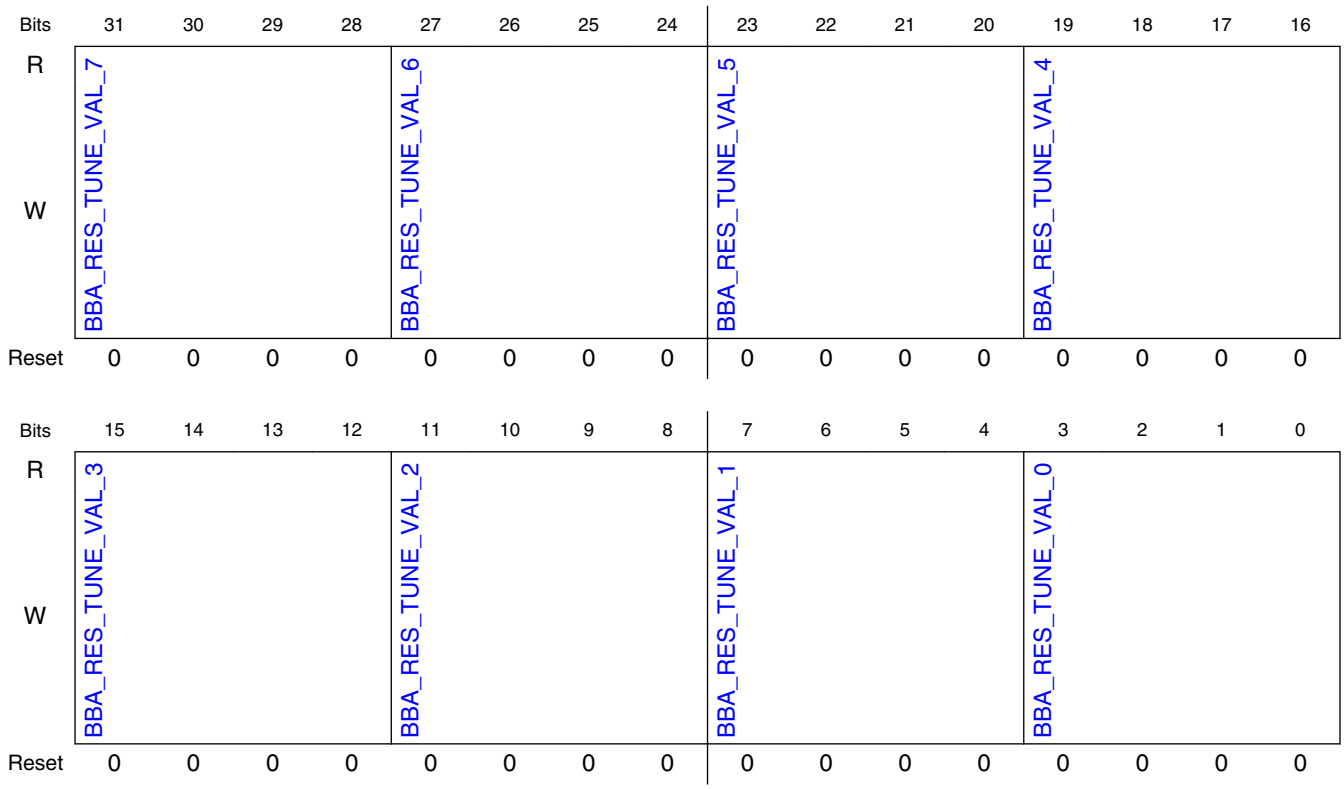
Field	Function
31-16 —	Reserved.
15-8 LNA_GAIN_VAL_9	LNA_GAIN step 9 Gain for LNA gain Step 9, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).
7-0 LNA_GAIN_VAL_8	LNA_GAIN step 8 Gain for LNA gain Step 8, used in RSSI calculation. Unsigned, 4*(measured_gain_dB).

A.2.3.23 BBA Resistor Tune Values 7..0 (BBA_RES_TUNE_VAL_7_0)

A.2.3.23.1 Offset

Register	Offset
BBA_RES_TUNE_VAL_7_0	5Ch

A.2.3.23.2 Diagram



A.2.3.23.3 Fields

Field	Function
31-28 BBA_RES_TUNE_VAL_7	BBA Resistor Tune Step 7 Gain offset for BBA gain Step 7, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
27-24 BBA_RES_TUNE_VAL_6	BBA Resistor Tune Step 6 Gain offset for BBA gain Step 6, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
23-20 BBA_RES_TUNE_VAL_5	BBA Resistor Tune Step 5 Gain offset for BBA gain Step 5, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
19-16 BBA_RES_TUNE_VAL_4	BBA Resistor Tune Step 4 Gain offset for BBA gain Step 4, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
15-12 BBA_RES_TUNE_VAL_3	BBA Resistor Tune Step 3 Gain offset for BBA gain Step 3, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
11-8	BBA Resistor Tune Step 2

Table continues on the next page...

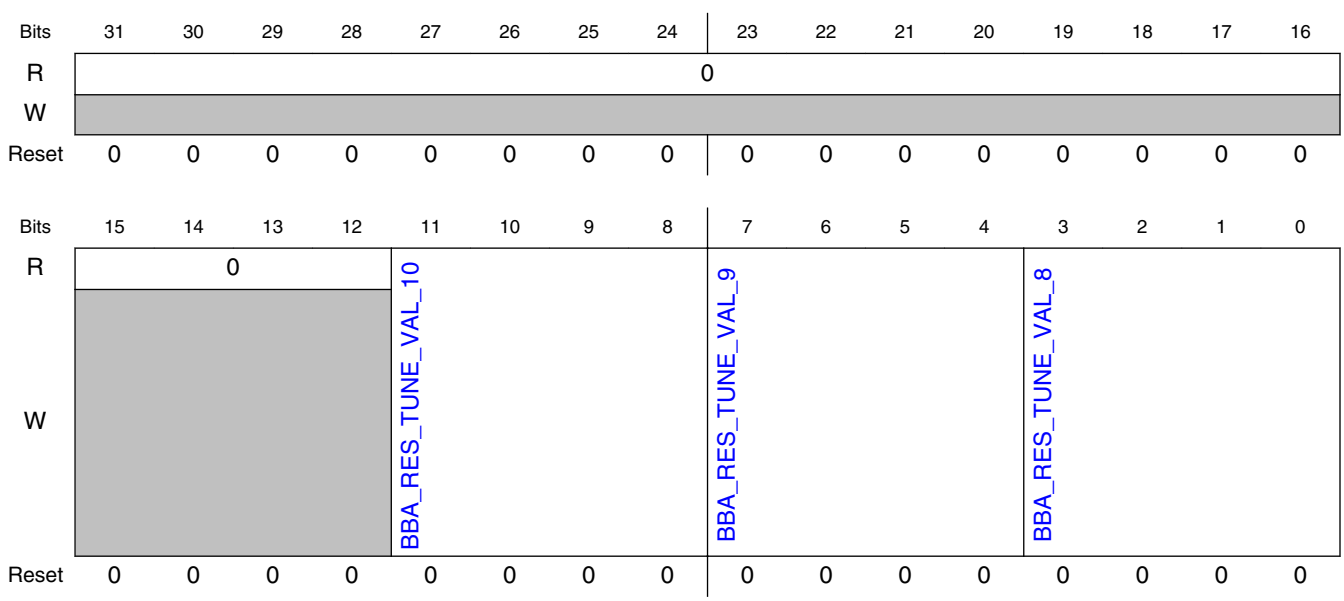
Field	Function
BBA_RES_TUNE_VAL_2	Gain offset for BBA gain Step 2, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
7-4 BBA_RES_TUNE_VAL_1	BBA Resistor Tune Step 1 Gain offset for BBA gain Step 1, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
3-0 BBA_RES_TUNE_VAL_0	BBA Resistor Tune Step 0 Gain offset for BBA gain Step 0, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).

A.2.3.24 BBA Resistor Tune Values 10..8 (BBA_RES_TUNE_VAL_10_8)

A.2.3.24.1 Offset

Register	Offset
BBA_RES_TUNE_VAL_10_8	60h

A.2.3.24.2 Diagram



A.2.3.24.3 Fields

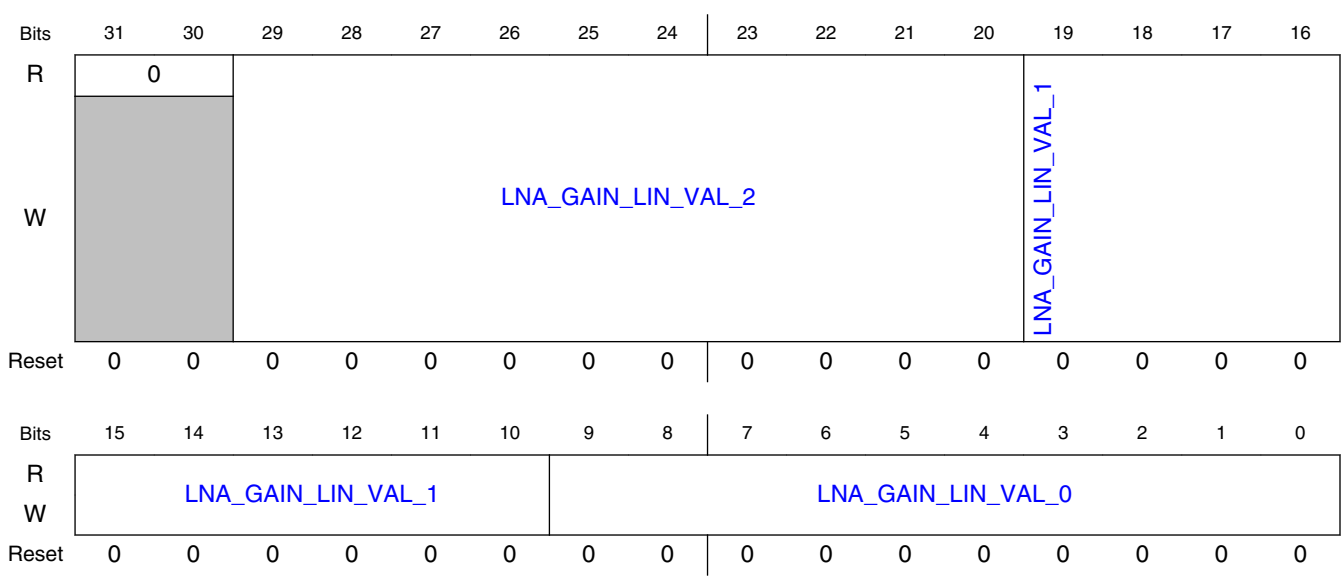
Field	Function
31-12 —	Reserved.
11-8 BBA_RES_TUNE_VAL_10	BBA Resistor Tune Step 10 Gain offset for BBA gain Step 10, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
7-4 BBA_RES_TUNE_VAL_9	BBA Resistor Tune Step 9 Gain offset for BBA gain Step 9, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).
3-0 BBA_RES_TUNE_VAL_8	BBA Resistor Tune Step 8 Gain offset for BBA gain Step 8, used in RSSI calculation. Signed, 2*(measured_gain_dB - spec_gain_dB).

A.2.3.25 LNA Linear Gain Values 2..0 (LNA_GAIN_LIN_VAL_2_0)

A.2.3.25.1 Offset

Register	Offset
LNA_GAIN_LIN_VAL_2_0	64h

A.2.3.25.2 Diagram



A.2.3.25.3 Fields

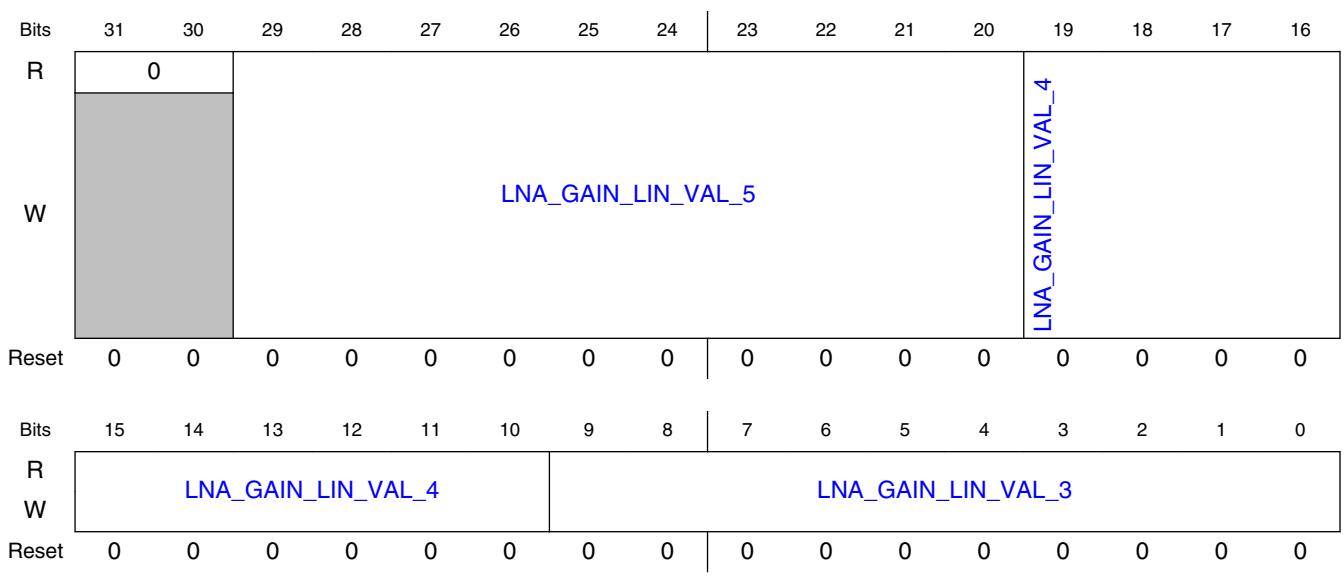
Field	Function
31-30 —	Reserved.
29-20 LNA_GAIN_LIN_VAL_2	LNA Linear Gain Step 2 LNA linear gain value for index 2, e.g. nominal value is 10^(2.2/20). Stored with 2 fractional bits, e.g. round([10^(2.2/20)]*2^2) = 5decimal. Format (8.2).
19-10 LNA_GAIN_LIN_VAL_1	LNA Linear Gain Step 1 LNA linear gain value for index 1, e.g. nominal value is 10^(-3.5/20). Stored with 2 fractional bits, e.g. round([10^(-3.5/20)]*2^2) = 3decimal. Format (8.2).
9-0 LNA_GAIN_LIN_VAL_0	LNA Linear Gain Step 0 LNA linear gain value for index 0, e.g. nominal value is 10^(-8.6/20). Stored with 2 fractional bits, e.g. round([10^(-8.6/20)]*2^2) = 1decimal. Format (8.2).

A.2.3.26 LNA Linear Gain Values 5..3 (LNA_GAIN_LIN_VAL_5_3)

A.2.3.26.1 Offset

Register	Offset
LNA_GAIN_LIN_VAL_5_3	68h

A.2.3.26.2 Diagram



A.2.3.26.3 Fields

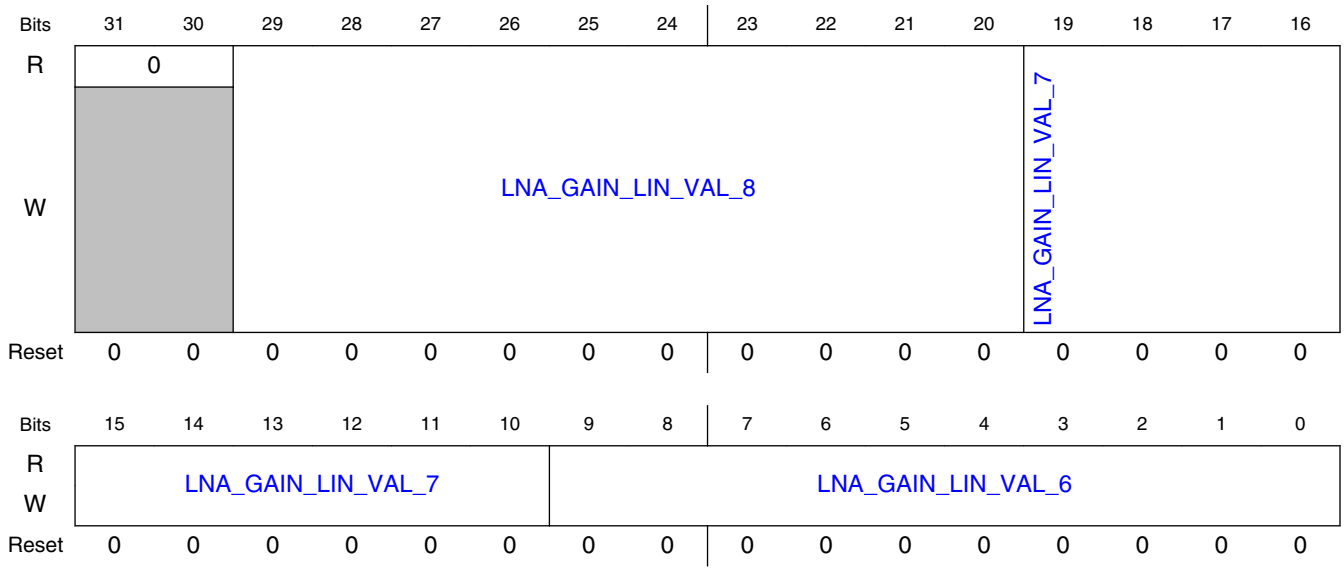
Field	Function
31-30 —	Reserved.
29-20 LNA_GAIN_LIN_VAL_5	LNA Linear Gain Step 5 LNA linear gain value for index 5, e.g. nominal value is 10^(22.7/20). Stored with 2 fractional bits, e.g. round([10^(22.7/20)]*2^2) = 55decimal. Format (8.2).
19-10 LNA_GAIN_LIN_VAL_4	LNA Linear Gain Step 4 LNA linear gain value for index 4, e.g. nominal value is 10^(19.8/20). Stored with 2 fractional bits, e.g. round([10^(19.8/20)]*2^2) = 39decimal. Format (8.2).
9-0 LNA_GAIN_LIN_VAL_3	LNA Linear Gain Step 3 LNA linear gain value for index 3, e.g. nominal value is 10^(13.9/20). Stored with 2 fractional bits, e.g. round([10^(13.9/20)]*2^2) = 20decimal. Format (8.2).

A.2.3.27 LNA Linear Gain Values 8..6 (LNA_GAIN_LIN_VAL_8_6)

A.2.3.27.1 Offset

Register	Offset
LNA_GAIN_LIN_VAL_8_6	6Ch

A.2.3.27.2 Diagram



A.2.3.27.3 Fields

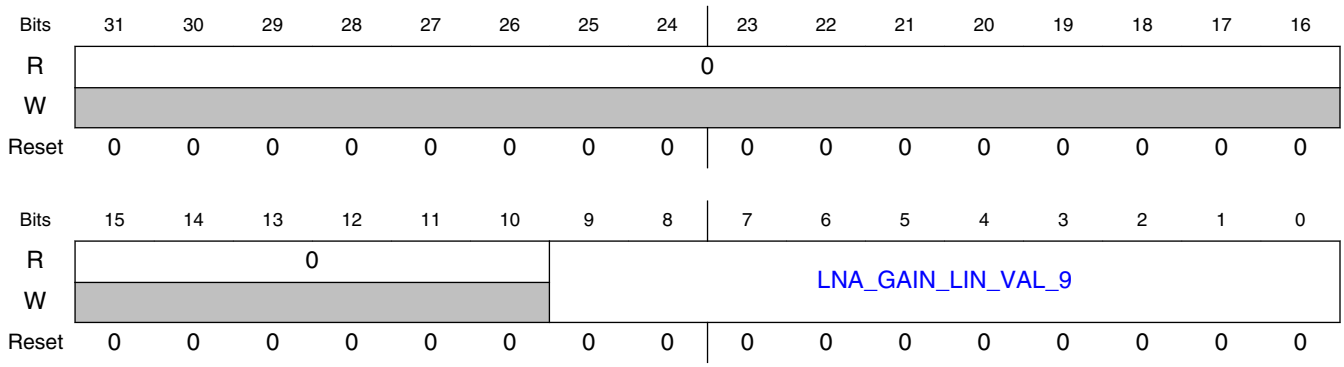
Field	Function
31-30 —	Reserved.
29-20 LNA_GAIN_LIN_VAL_8	LNA Linear Gain Step 8 LNA linear gain value for index 8, e.g. nominal value is 10^(39.9/20). Stored with 2 fractional bits, e.g. round([10^(39.9/20)]*2^2) = 396decimal. Format (8.2).
19-10 LNA_GAIN_LIN_VAL_7	LNA Linear Gain Step 7 LNA linear gain value for index 7, e.g. nominal value is 10^(34.4/20). Stored with 2 fractional bits, e.g. round([10^(34.4/20)]*2^2) = 210decimal. Format (8.2).
9-0 LNA_GAIN_LIN_VAL_6	LNA Linear Gain Step 6 LNA linear gain value for index 6, e.g. nominal value is 10^(28.6/20). Stored with 2 fractional bits, e.g. round([10^(28.6/20)]*2^2) = 108decimal. Format (8.2).

A.2.3.28 LNA Linear Gain Values 9 (LNA_GAIN_LIN_VAL_9)

A.2.3.28.1 Offset

Register	Offset
LNA_GAIN_LIN_VAL_9	70h

A.2.3.28.2 Diagram



A.2.3.28.3 Fields

Field	Function
31-10 —	Reserved.
9-0 LNA_GAIN_LIN_VAL_9	LNA Linear Gain Step 9 LNA linear gain value for index 9, e.g. nominal value is 10^(45.4/20). Stored with 2 fractional bits, e.g. round([10^(45.4/20)]*2^2) = 744decimal. Format (8.2).

A.2.3.29 BBA Resistor Tune Values 3..0 (BBA_RES_TUNE_LIN_VAL_3_0)

A.2.3.29.1 Offset

Register	Offset
BBA_RES_TUNE_LIN_VAL_3_0	74h

A.2.3.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BBA_RES_TUNE_LIN_VAL_3								BBA_RES_TUNE_LIN_VAL_2							
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	BBA_RES_TUNE_LIN_VAL_1								BBA_RES_TUNE_LIN_VAL_0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.29.3 Fields

Field	Function
31-24 BBA_RES_TUNE_LIN_VAL_3	BBA Resistor Tune Linear Gain Step 3 BBA linear gain value for index 3 (format: u5.3). Nominal value is 10^(9/20). Stored with 3 fractional bits, e.g. round([10^(9/20)]*2^3) = 23decimal
23-16 BBA_RES_TUNE_LIN_VAL_2	BBA Resistor Tune Linear Gain Step 2 BBA linear gain value for index 2 (format: u5.3). Nominal value is 10^(6/20). Stored with 3 fractional bits, e.g. round([10^(6/20)]*2^3) = 16decimal
15-8 BBA_RES_TUNE_LIN_VAL_1	BBA Resistor Tune Linear Gain Step 1 BBA linear gain value for index 1 (format: u5.3). Nominal value is 10^(3/20). Stored with 3 fractional bits, e.g. round([10^(3/20)]*2^3) = 11decimal
7-0 BBA_RES_TUNE_LIN_VAL_0	BBA Resistor Tune Linear Gain Step 0 BBA linear gain value for index 0 (format: u5.3). Nominal value is 10^(0/20). Stored with 3 fractional bits, e.g. round([10^(0/20)]*2^3) = 8decimal

A.2.3.30 BBA Resistor Tune Values 7..4 (BBA_RES_TUNE_LIN_VAL_7_4)

A.2.3.30.1 Offset

Register	Offset
BBA_RES_TUNE_LIN_VAL_7_4	78h

A.2.3.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BBA_RES_TUNE_LIN_VAL_7								BBA_RES_TUNE_LIN_VAL_6							
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	BBA_RES_TUNE_LIN_VAL_5								BBA_RES_TUNE_LIN_VAL_4							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.30.3 Fields

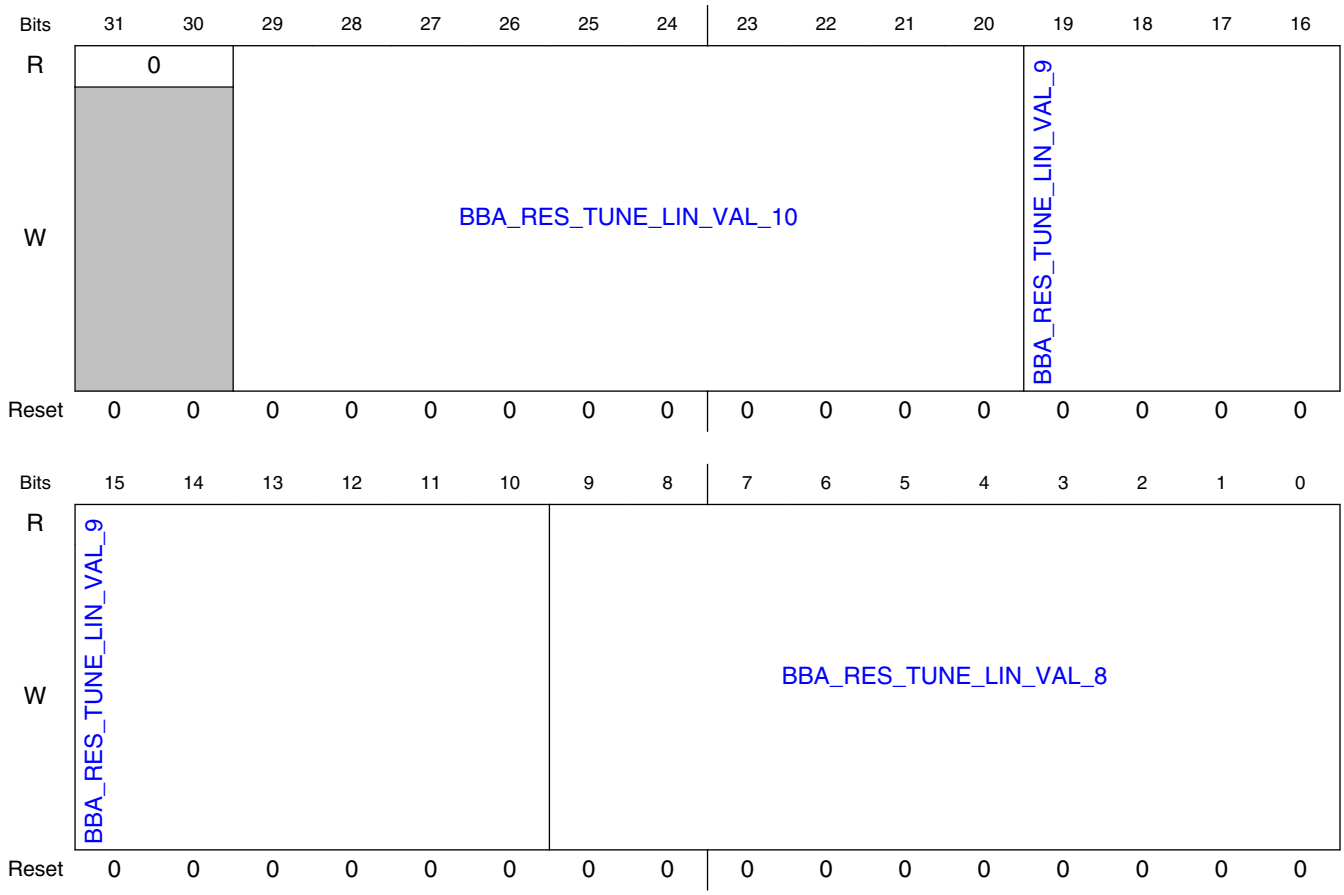
Field	Function
31-24 BBA_RES_TUNE_LIN_VAL_7	BBA Resistor Tune Linear Gain Step 7 BBA linear gain value for index 7 (format: u6.2). Nominal value is 10^(21/20). Stored with 2 fractional bits, e.g. round([10^(21/20)]*2^2) = 45 decimal
23-16 BBA_RES_TUNE_LIN_VAL_6	BBA Resistor Tune Linear Gain Step 6 BBA linear gain value for index 6 (format: u5.3). Nominal value is 10^(18/20). Stored with 3 fractional bits, e.g. round([10^(18/20)]*2^3) = 64decimal
15-8 BBA_RES_TUNE_LIN_VAL_5	BBA Resistor Tune Linear Gain Step 5 BBA linear gain value for index 5 (format: u5.3). Nominal value is 10^(15/20). Stored with 3 fractional bits, e.g. round([10^(15/20)]*2^3) = 45decimal
7-0 BBA_RES_TUNE_LIN_VAL_4	BBA Resistor Tune Linear Gain Step 4 BBA linear gain value for index 4 (format: u5.3). Nominal value is 10^(12/20). Stored with 3 fractional bits, e.g. round([10^(12/20)]*2^3) = 32decimal

A.2.3.31 BBA Resistor Tune Values 10..8 (BBA_RES_TUNE_LIN_VAL_10_8)

A.2.3.31.1 Offset

Register	Offset
BBA_RES_TUNE_LIN_VAL_10_8	7Ch

A.2.3.31.2 Diagram



A.2.3.31.3 Fields

Field	Function
31-30 —	Reserved.
29-20 BBA_RES_TUNE_LIN_VAL_10	BBA Resistor Tune Linear Gain Step 10 BBA linear gain value for index 10 (format: u7.3). Nominal value is 10^(30/20). Stored with 3 fractional bits, e.g. round([10^(30/20)]*2^3) = 253decimal
19-10 BBA_RES_TUNE_LIN_VAL_9	BBA Resistor Tune Linear Gain Step 9 BBA linear gain value for index 9 (format: u7.3). Nominal value is 10^(27/20). Stored with 3 fractional bits, e.g. round([10^(27/20)]*2^3) = 179decimal
9-0 BBA_RES_TUNE_LIN_VAL_8	BBA Resistor Tune Linear Gain Step 8 BBA linear gain value for index 8 (format: u7.3). Nominal value is 10^(24/20). Stored with 3 fractional bits, e.g. round([10^(24/20)]*2^3) = 127decimal

A.2.3.32 AGC Gain Tables Step 03..00 (AGC_GAIN_TBL_03_00)

A.2.3.32.1 Offset

Register	Offset
AGC_GAIN_TBL_03_00	80h

A.2.3.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_03				BBA_GAIN_03				LNA_GAIN_02				BBA_GAIN_02			
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	LNA_GAIN_01				BBA_GAIN_01				LNA_GAIN_00				BBA_GAIN_00			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.32.3 Fields

Field	Function
31-28 LNA_GAIN_03	LNA Gain 03 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24 BBA_GAIN_03	BBA Gain 03 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_02	LNA Gain 02 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_02	BBA Gain 02 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_01	LNA Gain 01 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_01	BBA Gain 01 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_00	LNA Gain 00 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_00	BBA Gain 00 BBA GAIN.

A.2.3.33 AGC Gain Tables Step 07..04 (AGC_GAIN_TBL_07_04)

A.2.3.33.1 Offset

Register	Offset
AGC_GAIN_TBL_07_04	84h

A.2.3.33.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_07				BBA_GAIN_07				LNA_GAIN_06				BBA_GAIN_06			
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	LNA_GAIN_05				BBA_GAIN_05				LNA_GAIN_04				BBA_GAIN_04			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.33.3 Fields

Field	Function
31-28 LNA_GAIN_07	LNA Gain 07 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24 BBA_GAIN_07	BBA Gain 07 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_06	LNA Gain 06 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_06	BBA Gain 06 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_05	LNA Gain 05 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_05	BBA Gain 05 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_04	LNA Gain 04 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0	BBA Gain 04

Field	Function
BBA_GAIN_04	BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

A.2.3.34 AGC Gain Tables Step 11..08 (AGC_GAIN_TBL_11_08)

A.2.3.34.1 Offset

Register	Offset
AGC_GAIN_TBL_11_08	88h

A.2.3.34.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_11				BBA_GAIN_11				LNA_GAIN_10				BBA_GAIN_10			
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	LNA_GAIN_09				BBA_GAIN_09				LNA_GAIN_08				BBA_GAIN_08			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.34.3 Fields

Field	Function
31-28 LNA_GAIN_11	LNA Gain 11 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24 BBA_GAIN_11	BBA Gain 11 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_10	LNA Gain 10 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_10	BBA Gain 10 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_09	LNA Gain 09 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8	BBA Gain 09

Table continues on the next page...

Field	Function
BBA_GAIN_09	BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_08	LNA Gain 08 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_08	BBA Gain 08 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

A.2.3.35 AGC Gain Tables Step 15..12 (AGC_GAIN_TBL_15_12)

A.2.3.35.1 Offset

Register	Offset
AGC_GAIN_TBL_15_12	8Ch

A.2.3.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_15				BBA_GAIN_15				LNA_GAIN_14				BBA_GAIN_14			
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	LNA_GAIN_13				BBA_GAIN_13				LNA_GAIN_12				BBA_GAIN_12			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.35.3 Fields

Field	Function
31-28 LNA_GAIN_15	LNA Gain 15 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24 BBA_GAIN_15	BBA Gain 15 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_14	LNA Gain 14 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16	BBA Gain 14

Table continues on the next page...

Field	Function
BBA_GAIN_14	BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_13	LNA Gain 13 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_13	BBA Gain 13 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_12	LNA Gain 12 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_12	BBA Gain 12 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

A.2.3.36 AGC Gain Tables Step 19..16 (AGC_GAIN_TBL_19_16)

A.2.3.36.1 Offset

Register	Offset
AGC_GAIN_TBL_19_16	90h

A.2.3.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_19				BBA_GAIN_19				LNA_GAIN_18				BBA_GAIN_18			
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	LNA_GAIN_17				BBA_GAIN_17				LNA_GAIN_16				BBA_GAIN_16			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.36.3 Fields

Field	Function
31-28 LNA_GAIN_19	LNA Gain 19 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24	BBA Gain 193

Table continues on the next page...

Field	Function
BBA_GAIN_19	BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_18	LNA Gain 18 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_18	BBA Gain 18 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_17	LNA Gain 17 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_17	BBA Gain 17 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_16	LNA Gain 16 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_16	BBA Gain 16 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

A.2.3.37 AGC Gain Tables Step 23..20 (AGC_GAIN_TBL_23_20)

A.2.3.37.1 Offset

Register	Offset
AGC_GAIN_TBL_23_20	94h

A.2.3.37.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LNA_GAIN_23				BBA_GAIN_23				LNA_GAIN_22				BBA_GAIN_22			
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	LNA_GAIN_21				BBA_GAIN_21				LNA_GAIN_20				BBA_GAIN_20			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.37.3 Fields

Field	Function
31-28 LNA_GAIN_23	LNA Gain 23 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
27-24 BBA_GAIN_23	BBA Gain 23 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
23-20 LNA_GAIN_22	LNA Gain 22 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_22	BBA Gain 22 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_21	LNA Gain 21 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_21	BBA Gain 21 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_20	LNA Gain 20 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_20	BBA Gain 20 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

A.2.3.38 AGC Gain Tables Step 26..24 (AGC_GAIN_TBL_26_24)

A.2.3.38.1 Offset

Register	Offset
AGC_GAIN_TBL_26_24	98h

A.2.3.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								LNA_GAIN_26				BBA_GAIN_26			
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	LNA_GAIN_25				BBA_GAIN_25				LNA_GAIN_24				BBA_GAIN_24			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.38.3 Fields

Field	Function
31-24 —	Reserved.
23-20 LNA_GAIN_26	LNA Gain 26 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
19-16 BBA_GAIN_26	BBA Gain 26 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
15-12 LNA_GAIN_25	LNA Gain 25 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
11-8 BBA_GAIN_25	BBA Gain 25 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.
7-4 LNA_GAIN_24	LNA Gain 24 LNA GAIN. See Table 45-20 for the mapping of LNA gain indexes to gain values.
3-0 BBA_GAIN_24	BBA Gain 24 BBA GAIN 0=0 dB, 1=3 dB, 2=6 dB … A=30 dB.

A.2.3.39 DCOC Offset (DCOC_OFFSET_0 - DCOC_OFFSET_26)

A.2.3.39.1 Offset

For a = 0 to 26:

Register	Offset
DCOC_OFFSET_a	A0h + (a × 4h)

A.2.3.39.2 Function

DCOC Offset Registers. These registers can only be accessed when the radio oscillator clock is active.

A.2.3.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DCOC_TZA_OFFSET_Q								DCOC_TZA_OFFSET_I							
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	0		DCOC_BBA_OFFSET_Q						0		DCOC_BBA_OFFSET_I					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.39.4 Fields

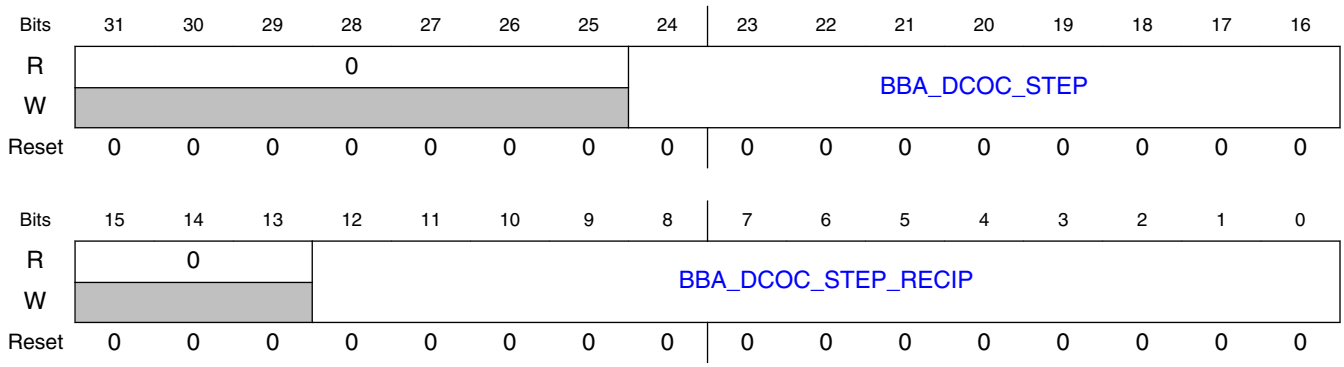
Field	Function
31-24 DCOC_TZA_OFFSET_Q	DCOC TZA Q-channel offset DCOC TZA Q-channel offset. When RX_DCOC_CAL_EN=1, this table is generated by the DCOC during calibration. When RX_DCOC_CAL_EN=0, this table may optionally be written by software.
23-16 DCOC_TZA_OFFSET_I	DCOC TZA I-channel offset DCOC TZA 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.
13-8 DCOC_BBA_OFFSET_Q	DCOC BBA Q-channel offset DCOC BBA 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.
7-6 —	Reserved.
5-0 DCOC_BBA_OFFSET_I	DCOC BBA I-channel offset DCOC BBA 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.

A.2.3.40 DCOC BBA DAC Step (DCOC_BBA_STEP)

A.2.3.40.1 Offset

Register	Offset
DCOC_BBA_STEP	10Ch

A.2.3.40.2 Diagram



A.2.3.40.3 Fields

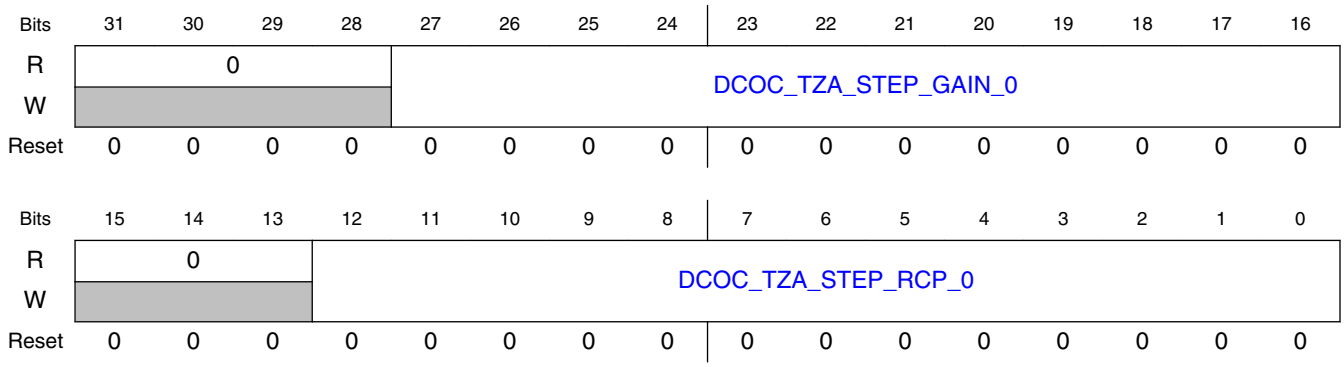
Field	Function
31-25 —	Reserved.
24-16 BBA_DCOC_STEP	DCOC BBA Step Size DCOC BBA Step Size (format: u6.3). This is the BBA DAC resolution in mV (1.2e3/32*63.44/120 = 19.83mV) times the ADC/decimator gain. This value is stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 BBA_DCOC_STEP_RECIP	DCOC BBA Reciprocal of Step Size DCOC BBA Reciprocal of Step Size (format: u.[00]13). This the reciprocal of the BBA DCOC STEP value. This value is stored as a 15 bit fraction (though only 13 bits are programmed).

A.2.3.41 DCOC TZA DAC Step 0 (DCOC_TZA_STEP_0)

A.2.3.41.1 Offset

Register	Offset
DCOC_TZA_STEP_0	110h

A.2.3.41.2 Diagram



A.2.3.41.3 Fields

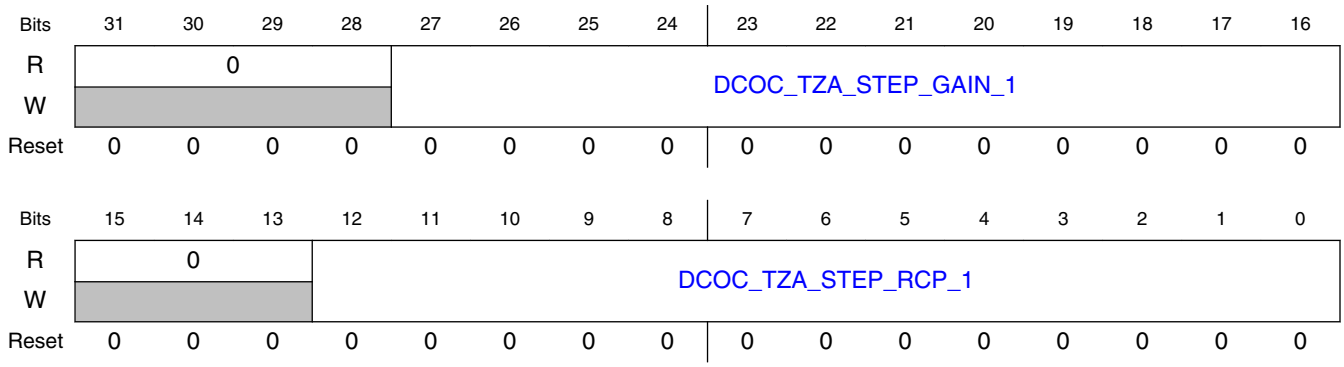
Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_0	DCOC_TZA_STEP_GAIN_0 DCOC TZA Step Size 0 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 0. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_0	DCOC_TZA_STEP_RCP_0 DCOC TZA Reciprocal of Step Size 0, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_0. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.42 DCOC TZA DAC Step 1 (DCOC_TZA_STEP_1)

A.2.3.42.1 Offset

Register	Offset
DCOC_TZA_STEP_1	114h

A.2.3.42.2 Diagram



A.2.3.42.3 Fields

Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_1	DCOC_TZA_STEP_GAIN_1 DCOC TZA Step Size 1 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 1. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_1	DCOC_TZA_STEP_RCP_1 DCOC TZA Reciprocal of Step Size 1, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_1. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.43 DCOC TZA DAC Step 2 (DCOC_TZA_STEP_2)

A.2.3.43.1 Offset

Register	Offset
DCOC_TZA_STEP_2	118h

A.2.3.43.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DCOC_TZA_STEP_GAIN_2											
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	0			DCOC_TZA_STEP_RCP_2												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.43.3 Fields

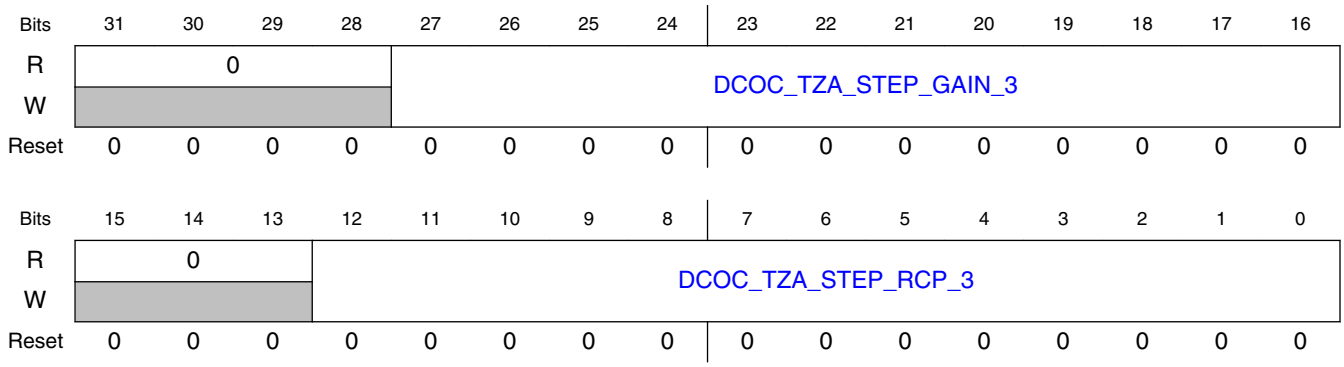
Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_2	DCOC_TZA_STEP_GAIN_2 DCOC TZA Step Size 2 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 2. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_2	DCOC_TZA_STEP_RCP_2 DCOC TZA Reciprocal of Step Size 2, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_2. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.44 DCOC TZA DAC Step 3 (DCOC_TZA_STEP_3)

A.2.3.44.1 Offset

Register	Offset
DCOC_TZA_STEP_3	11Ch

A.2.3.44.2 Diagram



A.2.3.44.3 Fields

Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_3	DCOC_TZA_STEP_GAIN_3 DCOC TZA Step Size 3 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 3. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_3	DCOC_TZA_STEP_RCP_3 DCOC TZA Reciprocal of Step Size 3, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_3. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.45 DCOC TZA DAC Step 4 (DCOC_TZA_STEP_4)

A.2.3.45.1 Offset

Register	Offset
DCOC_TZA_STEP_4	120h

A.2.3.45.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DCOC_TZA_STEP_GAIN_4											
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	0			DCOC_TZA_STEP_RCP_4												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.45.3 Fields

Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_4	DCOC_TZA_STEP_GAIN_4 DCOC TZA Step Size 4 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 4. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_4	DCOC_TZA_STEP_RCP_4 DCOC TZA Reciprocal of Step Size 4, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_4. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.46 DCOC TZA DAC Step 5 (DCOC_TZA_STEP_5)

A.2.3.46.1 Offset

Register	Offset
DCOC_TZA_STEP_5	124h

A.2.3.46.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DCOC_TZA_STEP_GAIN_5											
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	0			DCOC_TZA_STEP_RCP_5												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.46.3 Fields

Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_5	DCOC_TZA_STEP_GAIN_5 DCOC TZA Step Size 5 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 5. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_5	DCOC_TZA_STEP_RCP_5 DCOC TZA Reciprocal of Step Size 5, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_5. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.47 DCOC TZA DAC Step 6 (DCOC_TZA_STEP_6)

A.2.3.47.1 Offset

Register	Offset
DCOC_TZA_STEP_6	128h

A.2.3.47.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DCOC_TZA_STEP_GAIN_6											
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	0			DCOC_TZA_STEP_RCP_6												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.47.3 Fields

Field	Function
31-28 —	Reserved.
27-16 DCOC_TZA_STEP_GAIN_6	DCOC_TZA_STEP_GAIN_6 DCOC TZA Step Size 6 with gain (format u9.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 6. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_6	DCOC_TZA_STEP_RCP_6 DCOC TZA Reciprocal of Step Size 6, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_6. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.48 DCOC TZA DAC Step 7 (DCOC_TZA_STEP_7)

A.2.3.48.1 Offset

Register	Offset
DCOC_TZA_STEP_7	12Ch

A.2.3.48.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DCOC_TZA_STEP_GAIN_7												
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	0			DCOC_TZA_STEP_RCP_7												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.48.3 Fields

Field	Function
31-29 —	Reserved.
28-16 DCOC_TZA_STEP_GAIN_7	DCOC_TZA_STEP_GAIN_7 DCOC TZA Step Size 7 with gain (format u10.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 7. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_7	DCOC_TZA_STEP_RCP_7 DCOC TZA Reciprocal of Step Size 7, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_7. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.49 DCOC TZA DAC Step 5 (DCOC_TZA_STEP_8)

A.2.3.49.1 Offset

Register	Offset
DCOC_TZA_STEP_8	130h

A.2.3.49.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DCOC_TZA_STEP_GAIN_8												
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	0			DCOC_TZA_STEP_RCP_8												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.49.3 Fields

Field	Function
31-29 —	Reserved.
28-16 DCOC_TZA_STEP_GAIN_8	DCOC_TZA_STEP_GAIN_8 DCOC TZA Step Size 8 with gain (format u10.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 8. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_8	DCOC_TZA_STEP_RCP_8 DCOC TZA Reciprocal of Step Size 8, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_8. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.50 DCOC TZA DAC Step 9 (DCOC_TZA_STEP_9)

A.2.3.50.1 Offset

Register	Offset
DCOC_TZA_STEP_9	134h

A.2.3.50.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DCOC_TZA_STEP_GAIN_9													
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	0		DCOC_TZA_STEP_RCP_9													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.50.3 Fields

Field	Function
31-30 —	Reserved.
29-16 DCOC_TZA_STEP_GAIN_9	DCOC_TZA_STEP_GAIN_9 DCOC TZA Step Size 9 with gain (format u11.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 9. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_9	DCOC_TZA_STEP_RCP_9 DCOC TZA Reciprocal of Step Size 9, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_9. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.51 DCOC TZA DAC Step 10 (DCOC_TZA_STEP_10)

A.2.3.51.1 Offset

Register	Offset
DCOC_TZA_STEP_10	138h

A.2.3.51.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DCOC_TZA_STEP_GAIN_10													
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	0		DCOC_TZA_STEP_RCP_10													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.51.3 Fields

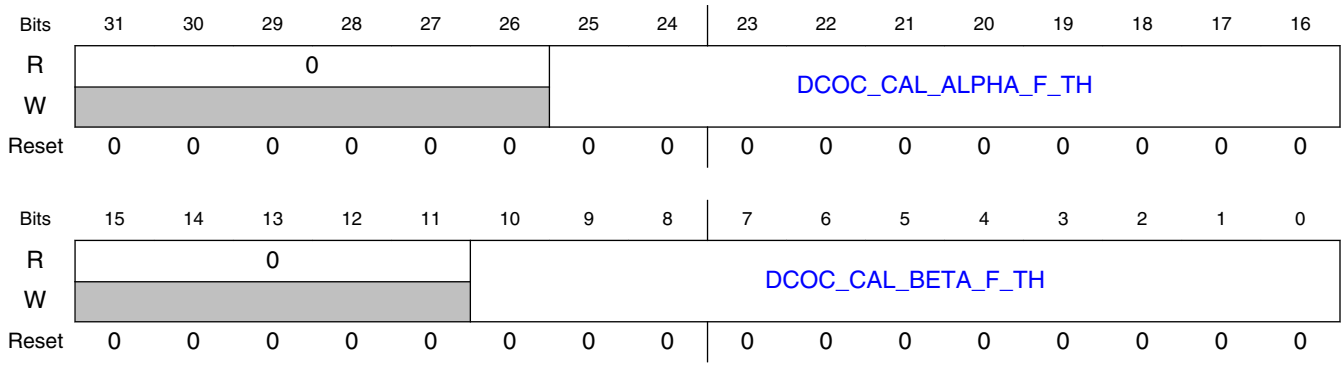
Field	Function
31-30 —	Reserved.
29-16 DCOC_TZA_STEP_GAIN_10	DCOC_TZA_STEP_GAIN_10 DCOC TZA Step Size 10 with gain (format u11.3). This is the TZA DAC resolution in mV (1.2e3/128*22.9/34 = 6.31mV) times the ADC/decimator gain times BBA gain for BBA index 10. This value iis stored in the register with 3 fractional bits.
15-13 —	Reserved.
12-0 DCOC_TZA_STEP_RCP_10	DCOC_TZA_STEP_RCP_10 DCOC TZA Reciprocal of Step Size 10, (format: u.[00]13). This the reciprocal of the DCOC_TZA_STEP_GAIN_10. The value is stored as a 15bit fractional value (though only 13 bits are programmed).

A.2.3.52 DCOC Calibration Fail Thresholds (DCOC_CAL_FAIL_TH)

A.2.3.52.1 Offset

Register	Offset
DCOC_CAL_FAIL_TH	160h

A.2.3.52.2 Diagram



A.2.3.52.3 Fields

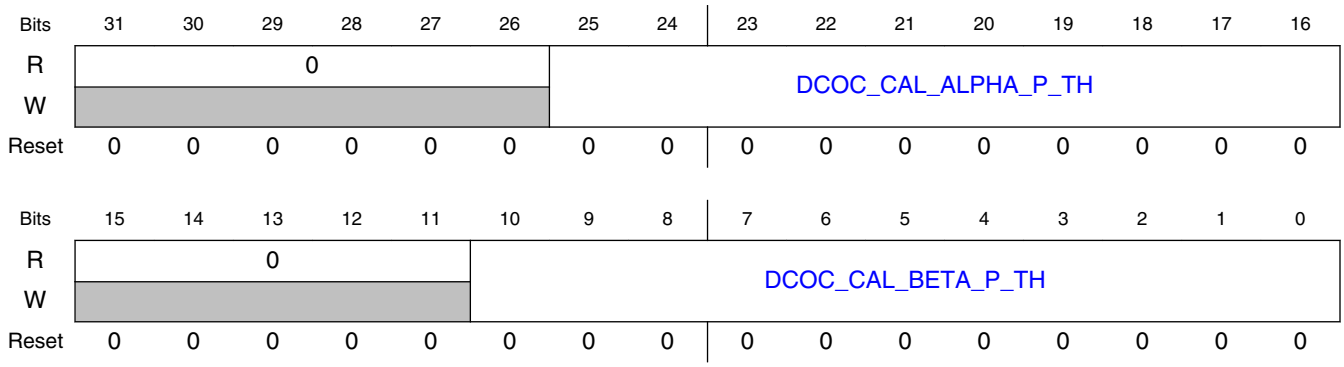
Field	Function
31-26 —	Reserved.
25-16 DCOC_CAL_ALPHA_F_TH	DCOC Calibration Alpha Fail Threshold If DCOC_CTRL_0[DCOC_CHECK_EN] is set, the absolute value of the alpha-hat I and Q estimates are compared gainst this fail threshold. If any of the estimates are greater than the fail threshold, the calibration is considered a failure. Format is u4.6.
15-11 —	Reserved.
10-0 DCOC_CAL_BETA_F_TH	DCOC Calibration Beta Fail Threshold If DCOC_CTRL_0[DCOC_CHECK_EN] is set, the absolute value of the beta-hat I and Q estimates are compared against this fail threshold. If any of the estimates are greater than the fail threshold, the calibration is considered a failure. Format is u11.

A.2.3.53 DCOC Calibration Pass Thresholds (DCOC_CAL_PASS_TH)

A.2.3.53.1 Offset

Register	Offset
DCOC_CAL_PASS_TH	164h

A.2.3.53.2 Diagram



A.2.3.53.3 Fields

Field	Function
31-26 —	Reserved.
25-16 DCOC_CAL_ALPHA_P_TH	DCOC Calibration Alpha Pass Threshold If DCOC_CTRL_0[DCOC_CHECK_EN] is set, the absolute value of the alpha-hat I and Q estimates are compared against this pass threshold. If all of the estimates are less than the pass threshold, the calibration is considered as passing. Format is u4.6.
15-11 —	Reserved.
10-0 DCOC_CAL_BETA_P_TH	DCOC Calibration Beta Pass Threshold If DCOC_CTRL_0[DCOC_CHECK_EN] is set, the absolute value of the beta-hat I and Q estimates are compared against this pass threshold. If all of the estimates are less than the pass threshold, the calibration is considered as passing. Format is u11.

A.2.3.54 DCOC Calibration Alpha (DCOC_CAL_ALPHA)

A.2.3.54.1 Offset

Register	Offset
DCOC_CAL_ALPHA	168h

A.2.3.54.2 Diagram

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

A.2.3.54.3 Fields

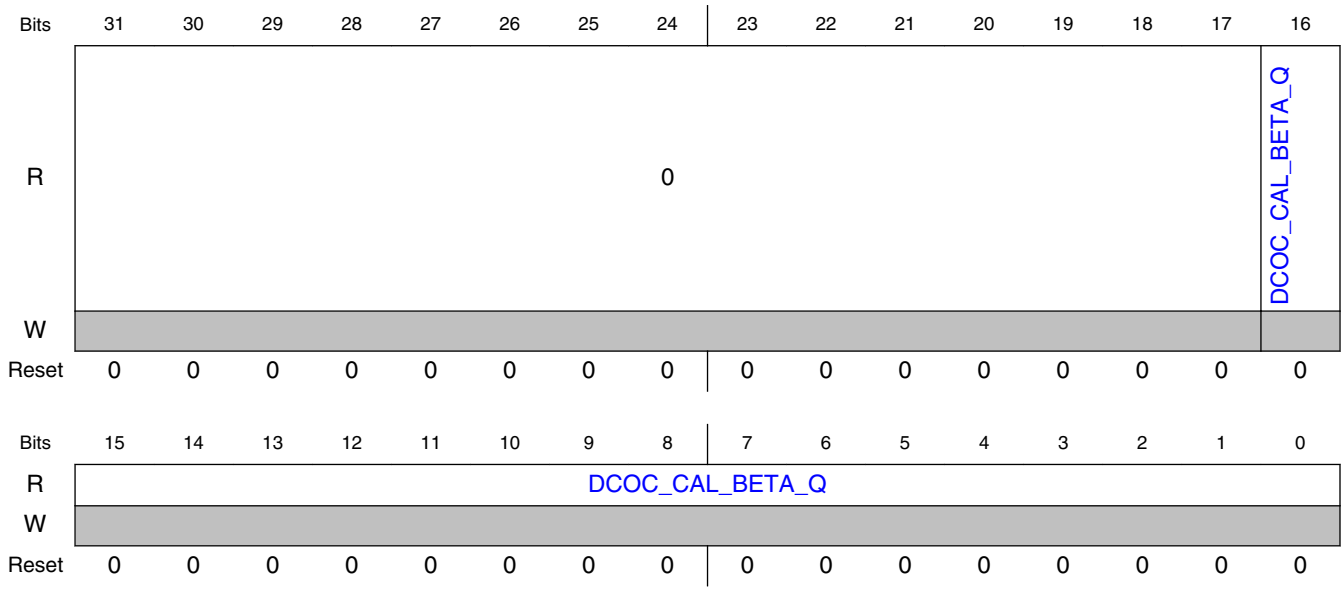
Field	Function
31-27 —	Reserved.
26-16 DCOC_CAL_ALPHA_Q	DCOC_CAL_ALPHA_Q DCOC Calibration Q-channel ALPHA. This read-only value represents the Q channel estimate of the ALPHA DC component calculated in DCOC calibration. Format s4.6. This is provided for debug/ characterization purposes.
15-11 —	Reserved.
10-0 DCOC_CAL_ALPHA_I	DCOC Calibration I-channel ALPHA constant DCOC Calibration I-channel ALPHA. This read-only value represents the I channel estimate of the ALPHA DC component calculated in DCOC calibration. Format s4.6. This is provided for debug/ characterization purposes.

A.2.3.55 DCOC Calibration Beta Q (DCOC_CAL_BETA_Q)

A.2.3.55.1 Offset

Register	Offset
DCOC_CAL_BETA_Q	16Ch

A.2.3.55.2 Diagram



A.2.3.55.3 Fields

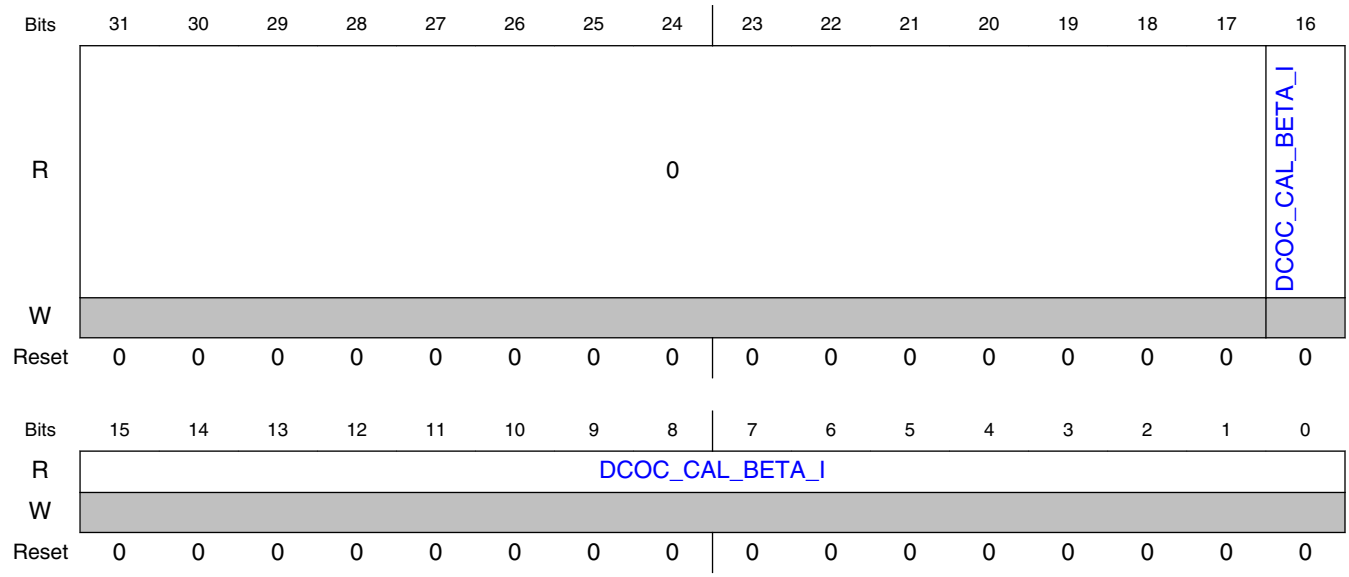
Field	Function
31-17 —	Reserved.
16-0 DCOC_CAL_BETA_Q	DCOC_CAL_BETA_Q DCOC Calibration Q-channel BETA. This read-only value represents the Q channel estimate of the BETA DC component calculated in DCOC calibration. Format s11.5. This is provided for debug/characterization purposes.

A.2.3.56 DCOC Calibration Beta I (DCOC_CAL_BETA_I)

A.2.3.56.1 Offset

Register	Offset
DCOC_CAL_BETA_I	170h

A.2.3.56.2 Diagram



A.2.3.56.3 Fields

Field	Function
31-17 —	Reserved.
16-0 DCOC_CAL_BETA_I	DCOC_CAL_BETA_I DCOC Calibration I-channel BETA. This read-only value represents the I channel estimate of the BETA DC component calculated in DCOC calibration. Format s11.5. This is provided for debug/characterization purposes.

A.2.3.57 DCOC Calibration Gamma (DCOC_CAL_GAMMA)

A.2.3.57.1 Offset

Register	Offset
DCOC_CAL_GAMMA	174h

A.2.3.57.2 Diagram

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

A.2.3.57.3 Fields

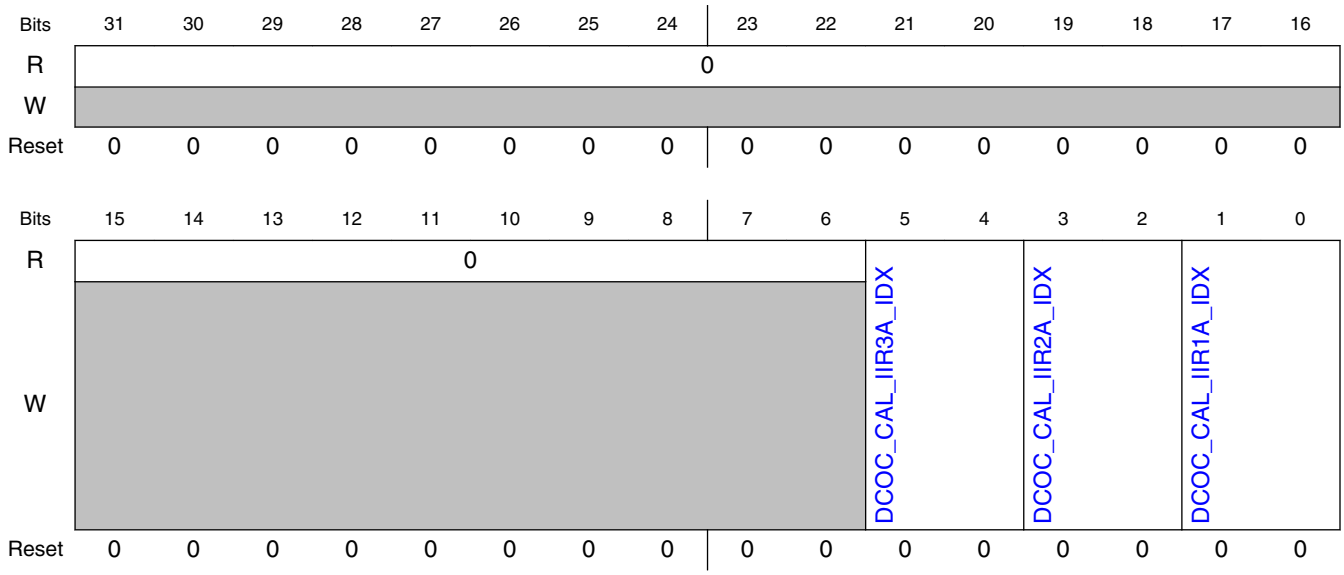
Field	Function
31-16 DCOC_CAL_GAMMA_Q	DCOC_CAL_GAMMA_Q DCOC Calibration Q-channel GAMMA. This read-only value represents the Q channel estimate of the GAMMA DC component calculated in DCOC calibration. Format s11.4. This is provided for debug/ characterization purposes.
15-0 DCOC_CAL_GAMMA_I	DCOC_CAL_GAMMA_I DCOC Calibration I-channel GAMMA. This read-only value represents the I channel estimate of the GAMMA DC component calculated in DCOC calibration. Format s11.4. This is provided for debug/ characterization purposes.

A.2.3.58 DCOC Calibration IIR (DCOC_CAL_IIR)

A.2.3.58.1 Offset

Register	Offset
DCOC_CAL_IIR	178h

A.2.3.58.2 Diagram



A.2.3.58.3 Fields

Field	Function
31-6 —	Reserved.
5-4 DCOC_CAL_IIR 3A_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. 00b - 1/4 01b - 1/8 10b - 1/16 11b - 1/32
3-2 DCOC_CAL_IIR 2A_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. 00b - 1/1 01b - 1/4 10b - 1/8 11b - 1/16
1-0 DCOC_CAL_IIR 1A_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. 00b - 1/1 01b - 1/4 10b - 1/8 11b - 1/16

A.2.3.59 DCOC Calibration Result (DCOC_CAL1 - DCOC_CAL3)

A.2.3.59.1 Offset

For a = 1 to 3:

Register	Offset
DCOC_CALa	17Ch + (a × 4h)

A.2.3.59.2 Function

Result of one of the calibration iterations.

A.2.3.59.3 Diagram

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

A.2.3.59.4 Fields

Field	Function
31-28 —	Reserved.
27-16 DCOC_CAL_RES_Q	DCOC Calibration Result - Q Channel Q channel DCOC calibration result. This value represents the DCOC's Q channel DC estimate for the nth calibration gain setting. Format s11. This is provided for debug/chacterization purposes.
15-12 —	Reserved.
11-0 DCOC_CAL_RES_I	DCOC Calibration Result - I Channel I channel DCOC calibration result. This value represents the DCOC's I channel DC estimate for the nth calibration gain setting. Format s11. This is provided for debug/chacterization purposes.

A.2.3.60 RX_DIG CCA ED LQI Control Register 0 (CCA_ED_LQI_CTRL_0)

A.2.3.60.1 Offset

Register	Offset
CCA_ED_LQI_CTRL_0	190h

A.2.3.60.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		SNR_ADJ						LQI_CNTR							
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	CORR_CNTR_THRESH								LQI_CORR_THRESH							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.60.3 Fields

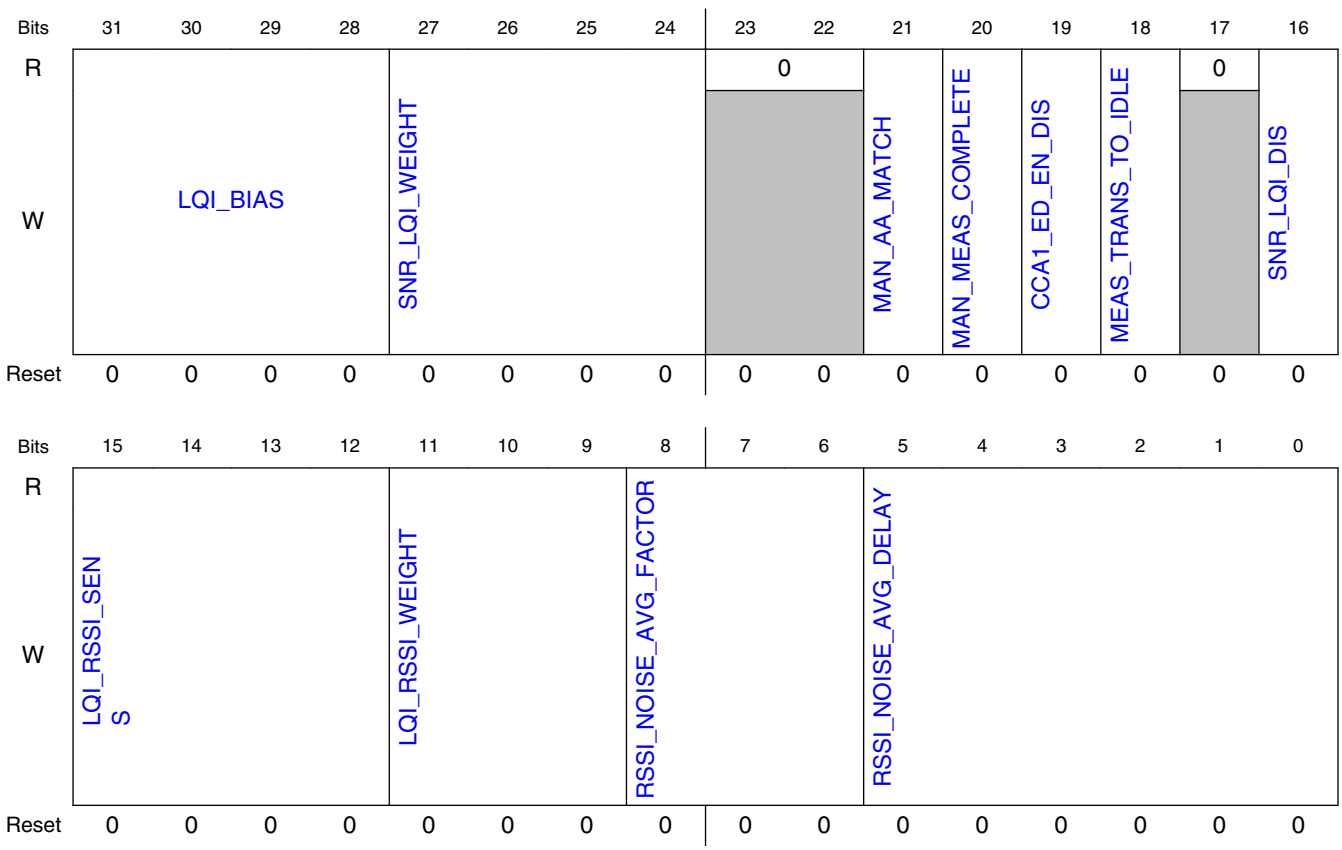
Field	Function
31-30 —	Reserved.
29-24 SNR_ADJ	SNR calculation adjustment Signed value with quarter dB resolution.
23-16 LQI_CNTR	LQI Counter Duration of LQI in uS.
15-8 CORR_CNTR_THRESH	Correlation Count Threshold Threshold used to compare the counted correlation magnitudes exceeding LQI_CORR_THRESH.
7-0 LQI_CORR_THRESH	LQI Correlation Threshold Threshold used to compare correlation magnitude values from the PHY

A.2.3.61 RX_DIG CCA ED LQI Control Register 1 (CCA_ED_LQI_CTRL_1)

A.2.3.61.1 Offset

Register	Offset
CCA_ED_LQI_CTRL_1	194h

A.2.3.61.2 Diagram



A.2.3.61.3 Fields

Field	Function
31-28 LQI_BIAS	LQI Bias. Bias used for LQI calculation. Applied bias = -36 + 2*LQI_BIAS.
27-24	SNR LQI Weight SNR weight used for LQI calculation

Table continues on the next page...

Field	Function
SNR_LQI_WEIGHT	0000b - 0.0 0001b - 1.0 0010b - 1.125 0011b - 1.25 0100b - 1.375 0101b - 1.5 0110b - 1.625 0111b - 1.75 1000b - 1.875 1001b - 2.0 1010b - 2.125 1011b - 2.25 1100b - 2.375 1101b - 2.5 1110b - 2.625 1111b - 2.75
23-22 —	Reserved.
21 MAN_AA_MATCH	Manual AA Match When set, this causes an AA match condition. Intended to be used only for debug. 0b - Normal operation 1b - Manually asserts the AA match signal for the RX_DIG CCA/ED/LQI and AGC blocks. Intended to be used only for debug.
20 MAN_MEAS_COMPLETE	Manual measurement complete 0b - Normal operation 1b - Manually asserts the measurement complete signal for the RX_DIG CCA/ED/LQI blocks. Intended to be used only for debug.
19 CCA1_ED_EN_DIS	CCA1_ED_EN Disable 0b - Normal operation 1b - CCA1_ED_EN input is disabled
18 MEAS_TRANS_TO_IDLE	Measurement Transition to IDLE Establishes the state machine transition following an LQI or CCA1/ED measurement 0b - Module transitions to RSSI state 1b - Module transitions to IDLE state
17 —	Reserved.
16 SNR_LQI_DIS	SNR LQI Disable 0b - Normal operation. 1b - The RX_DIG CCA/ED/LQI block ignores the AA match input which starts an LQI measurement.
15-12 LQI_RSSI_SENS	LQI RSSI Sensitivity Unsigned integer used for calculation of LQI sensitivity. Sensitivity = -103 + LQI_RSSI_SENS.
11-9 LQI_RSSI_WEIGHT	LQI RSSI Weight RSSI weight used for LQI calculation 000b - 2.0 001b - 2.125 010b - 2.25 011b - 2.375 100b - 2.5

Table continues on the next page...

Field	Function
	101b - 2.625 110b - 2.75 111b - 2.875
8-6 RSSI_NOISE_AVG_FACTOR	RSSI Noise Averaging Factor Factor used for RSSI and SNR averaging. 000b - 1 001b - 64 010b - 70 011b - 128 100b - 139 101b - 256 110b - 277 111b - 512
5-0 RSSI_NOISE_AVG_DELAY	RSSI Noise Averaging Delay Programmable delay used to delay the enable of averagers "RSSI Averager" and "Noise Averager" after a cca1_ed_trig assertion (in wideband mode) or after a aa_sfd_matched assertion (in narrowband mode). The delay is expressed in ticks of frequenc Fs/N, where Fs is the ADC decimator output sampling frequency in wideband mode or the base-band sampling frequency in the narrowband mode.

A.2.3.62 RX_DIG CCA ED LQI Status Register 0 (CCA_ED_LQI_STAT_0)

A.2.3.62.1 Offset

Register	Offset
CCA_ED_LQI_STAT_0	198h

A.2.3.62.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						MEAS_COMPLETE	CCA1_STATE	SNR_OUT							
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	ED_OUT								LQI_OUT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.62.3 Fields

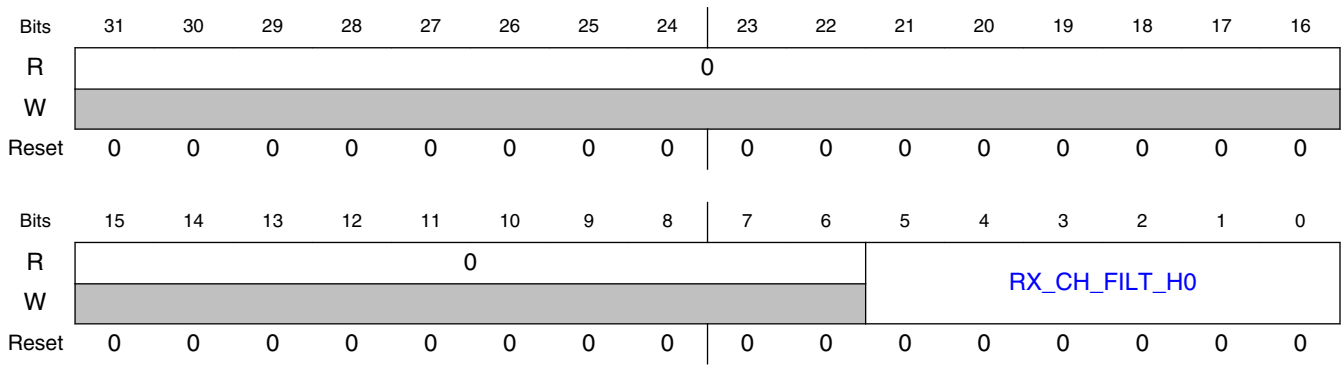
Field	Function
31-26 —	Reserved.
25 MEAS_COMPLETE	Measurement Complete Set upon measurement complete
24 CCA1_STATE	CCA1 State Reflects the state of CCA1 channel state
23-16 SNR_OUT	SNR output Reflects the SNR measurement value.
15-8 ED_OUT	ED output Reflects the Energy Detect (DC) measurement value.
7-0 LQI_OUT	LQI output Reflects the LQI measurement value.

A.2.3.63 Receive Channel Filter Coefficient 0 (RX_CHF_COEF_0)

A.2.3.63.1 Offset

Register	Offset
RX_CHF_COEF_0	1A0h

A.2.3.63.2 Diagram



A.2.3.63.3 Fields

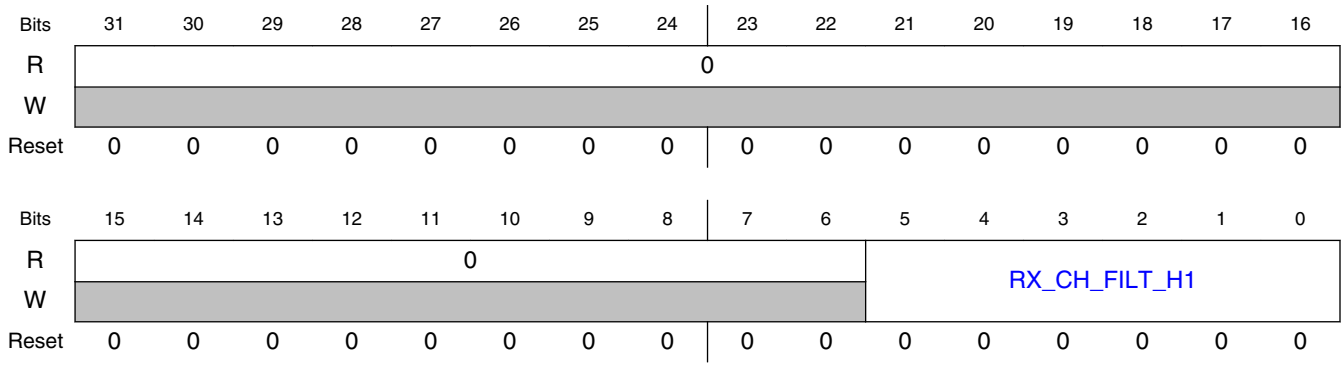
Field	Function
31-6 —	Reserved.
5-0 RX_CH_FILT_H0	RX Channel Filter Coefficient 0 RX Channel Filter Coefficient 0, 6-bit signed fractional

A.2.3.64 Receive Channel Filter Coefficient 1 (RX_CHF_COEF_1)

A.2.3.64.1 Offset

Register	Offset
RX_CHF_COEF_1	1A4h

A.2.3.64.2 Diagram



A.2.3.64.3 Fields

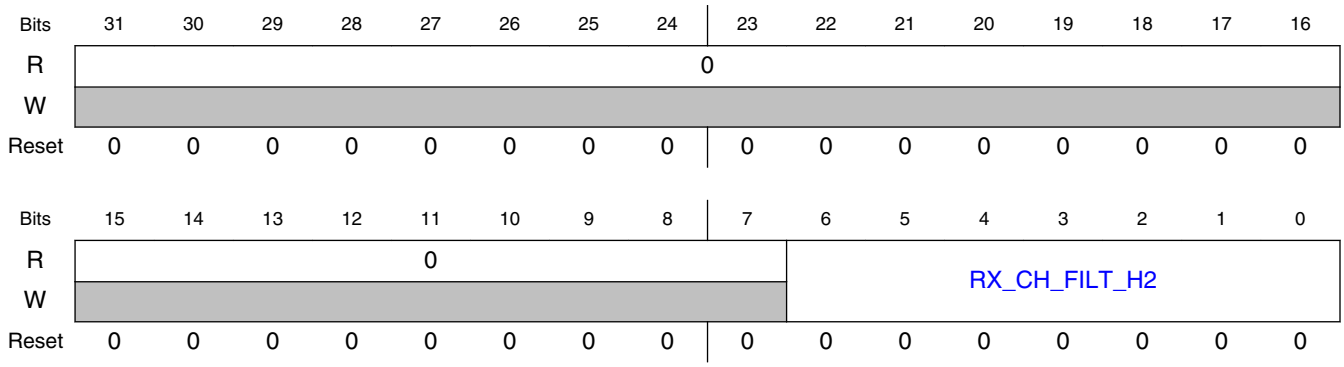
Field	Function
31-6 —	Reserved.
5-0 RX_CH_FILT_H1	RX Channel Filter Coefficient 1 RX Channel Filter Coefficient 1, 6-bit signed fractional

A.2.3.65 Receive Channel Filter Coefficient 2 (RX_CHF_COEF_2)

A.2.3.65.1 Offset

Register	Offset
RX_CHF_COEF_2	1A8h

A.2.3.65.2 Diagram



A.2.3.65.3 Fields

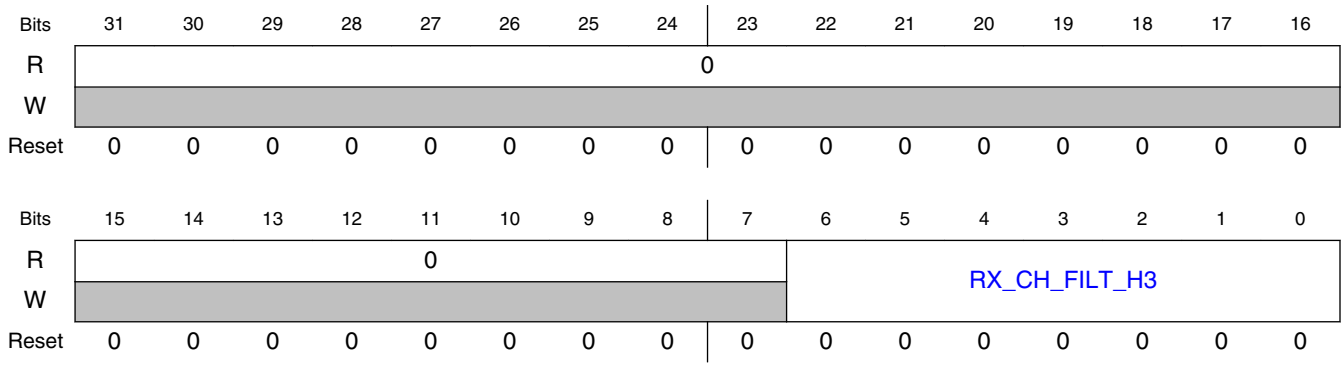
Field	Function
31-7 —	Reserved.
6-0 RX_CH_FILT_H2	RX Channel Filter Coefficient 2 RX Channel Filter Coefficient 2, 7-bit signed fractional

A.2.3.66 Receive Channel Filter Coefficient 3 (RX_CHF_COEF_3)

A.2.3.66.1 Offset

Register	Offset
RX_CHF_COEF_3	1ACh

A.2.3.66.2 Diagram



A.2.3.66.3 Fields

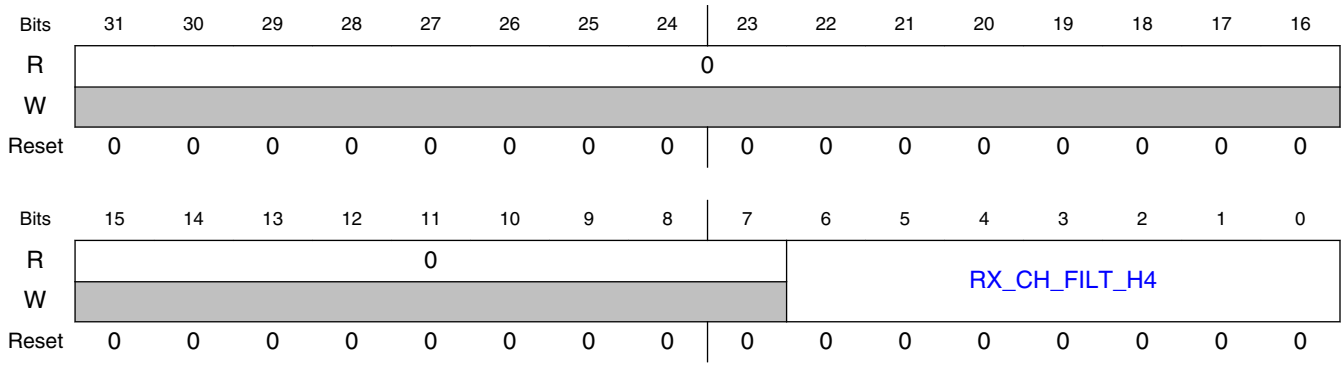
Field	Function
31-7 —	Reserved.
6-0 RX_CH_FILT_H3	RX Channel Filter Coefficient 3 RX Channel Filter Coefficient 3, 7-bit signed fractional

A.2.3.67 Receive Channel Filter Coefficient 4 (RX_CHF_COEF_4)

A.2.3.67.1 Offset

Register	Offset
RX_CHF_COEF_4	1B0h

A.2.3.67.2 Diagram



A.2.3.67.3 Fields

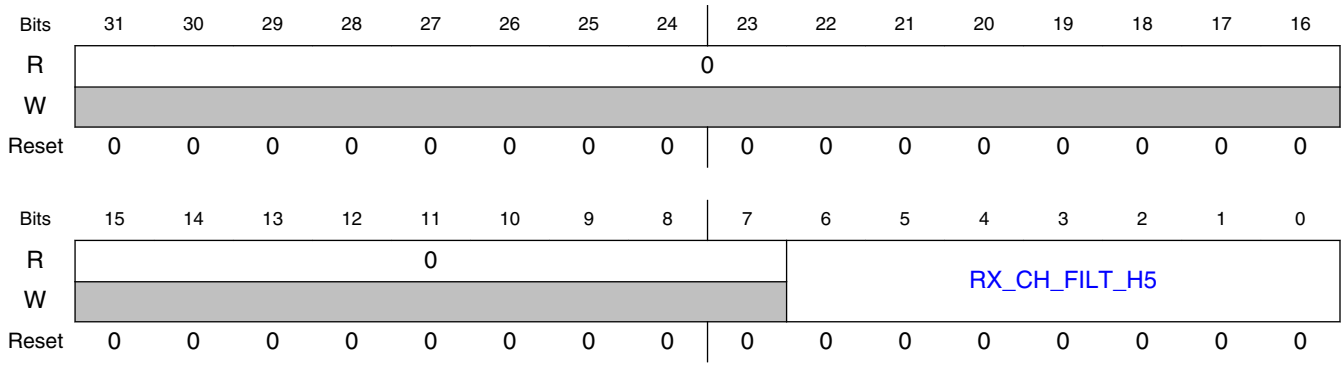
Field	Function
31-7 —	Reserved.
6-0 RX_CH_FILT_H4	RX Channel Filter Coefficient 4 RX Channel Filter Coefficient 4, 7-bit signed fractional

A.2.3.68 Receive Channel Filter Coefficient 5 (RX_CHF_COEF_5)

A.2.3.68.1 Offset

Register	Offset
RX_CHF_COEF_5	1B4h

A.2.3.68.2 Diagram



A.2.3.68.3 Fields

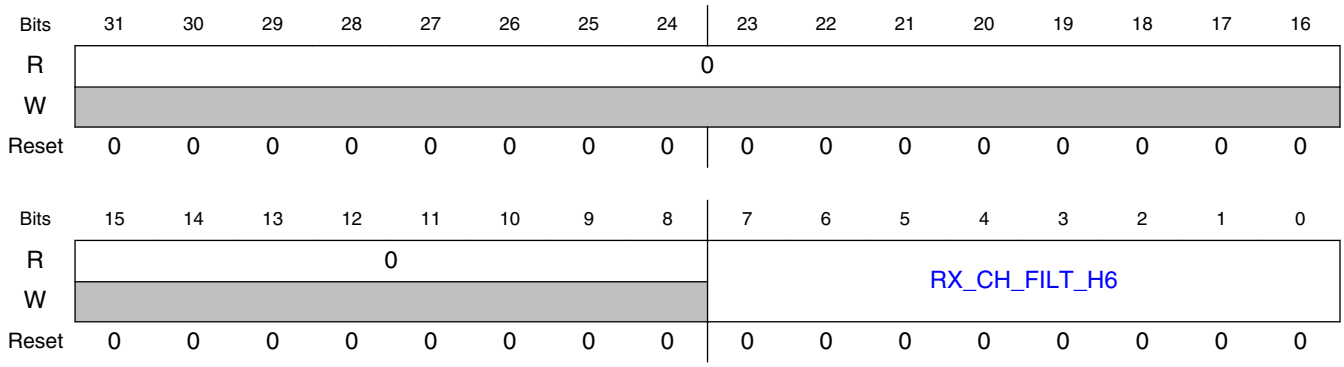
Field	Function
31-7 —	Reserved.
6-0 RX_CH_FILT_H5	RX Channel Filter Coefficient 5 RX Channel Filter Coefficient 5, 7-bit signed fractional

A.2.3.69 Receive Channel Filter Coefficient 6 (RX_CHF_COEF_6)

A.2.3.69.1 Offset

Register	Offset
RX_CHF_COEF_6	1B8h

A.2.3.69.2 Diagram



A.2.3.69.3 Fields

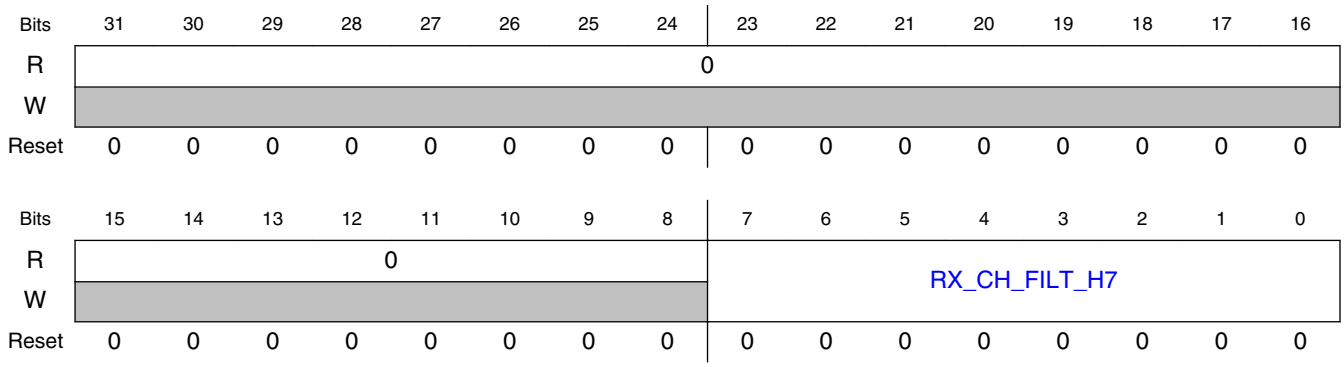
Field	Function
31-8 —	Reserved.
7-0 RX_CH_FILT_H6	RX Channel Filter Coefficient 6 RX Channel Filter Coefficient 6, 8-bit signed fractional

A.2.3.70 Receive Channel Filter Coefficient 7 (RX_CHF_COEF_7)

A.2.3.70.1 Offset

Register	Offset
RX_CHF_COEF_7	1BCh

A.2.3.70.2 Diagram



A.2.3.70.3 Fields

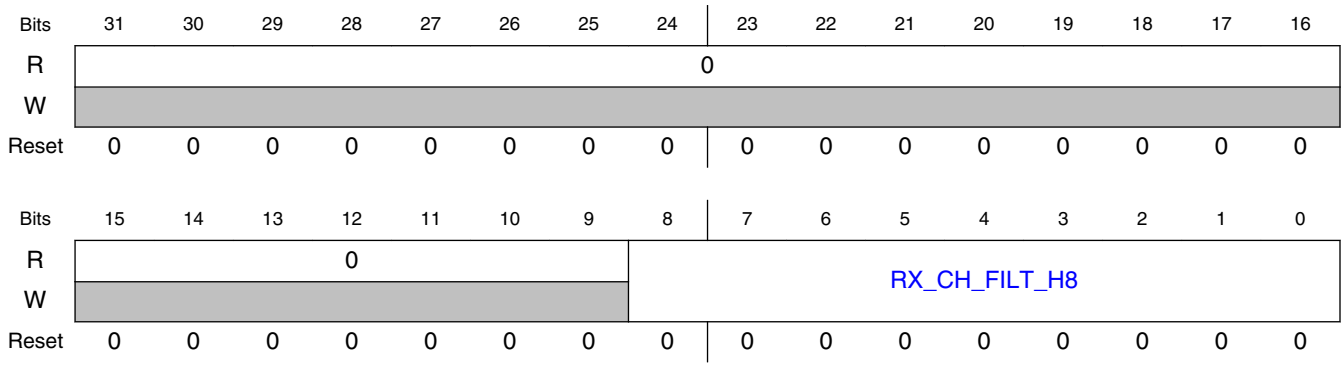
Field	Function
31-8 —	Reserved.
7-0 RX_CH_FILT_H7	RX Channel Filter Coefficient 7 RX Channel Filter Coefficient 7, 8-bit signed fractional

A.2.3.71 Receive Channel Filter Coefficient 8 (RX_CHF_COEF_8)

A.2.3.71.1 Offset

Register	Offset
RX_CHF_COEF_8	1C0h

A.2.3.71.2 Diagram



A.2.3.71.3 Fields

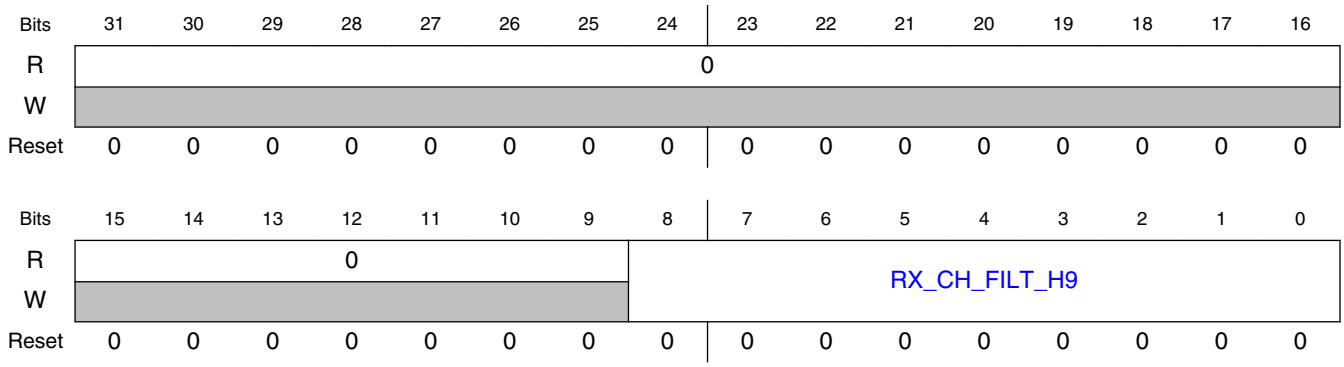
Field	Function
31-9 —	Reserved.
8-0 RX_CH_FILT_H8	RX Channel Filter Coefficient 8 RX Channel Filter Coefficient 8, 9-bit signed fractional

A.2.3.72 Receive Channel Filter Coefficient 9 (RX_CHF_COEF_9)

A.2.3.72.1 Offset

Register	Offset
RX_CHF_COEF_9	1C4h

A.2.3.72.2 Diagram



A.2.3.72.3 Fields

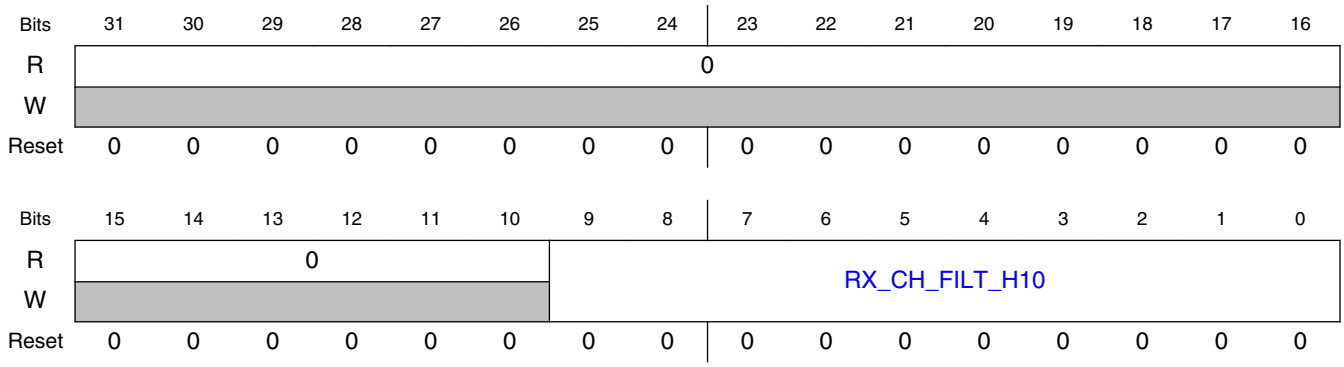
Field	Function
31-9 —	Reserved.
8-0 RX_CH_FILT_H9	RX Channel Filter Coefficient 9 RX Channel Filter Coefficient 9, 9-bit signed fractional

A.2.3.73 Receive Channel Filter Coefficient 10 (RX_CHF_COEF_10)

A.2.3.73.1 Offset

Register	Offset
RX_CHF_COEF_10	1C8h

A.2.3.73.2 Diagram



A.2.3.73.3 Fields

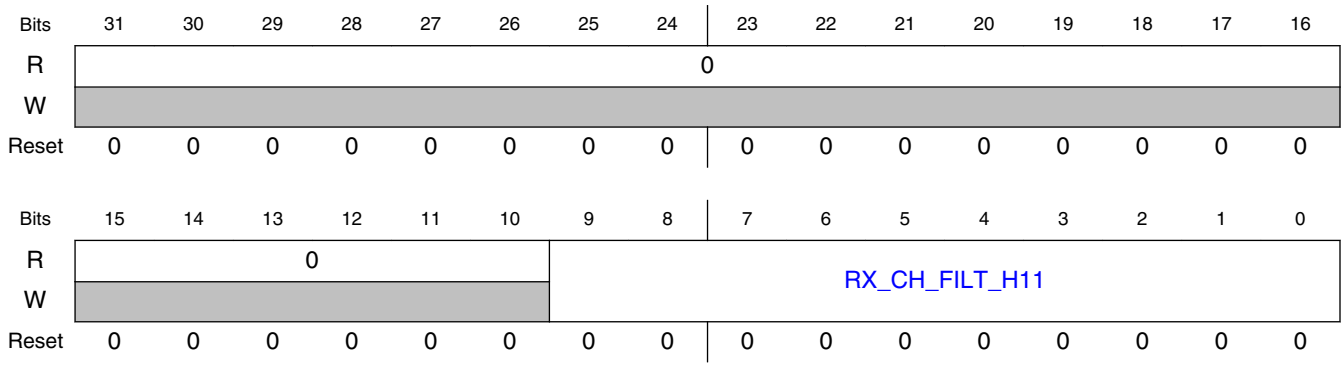
Field	Function
31-10 —	Reserved.
9-0 RX_CH_FILT_H10	RX Channel Filter Coefficient 10 RX Channel Filter Coefficient 10, 10-bit signed fractional

A.2.3.74 Receive Channel Filter Coefficient 11 (RX_CHF_COEF_11)

A.2.3.74.1 Offset

Register	Offset
RX_CHF_COEF_11	1CCh

A.2.3.74.2 Diagram



A.2.3.74.3 Fields

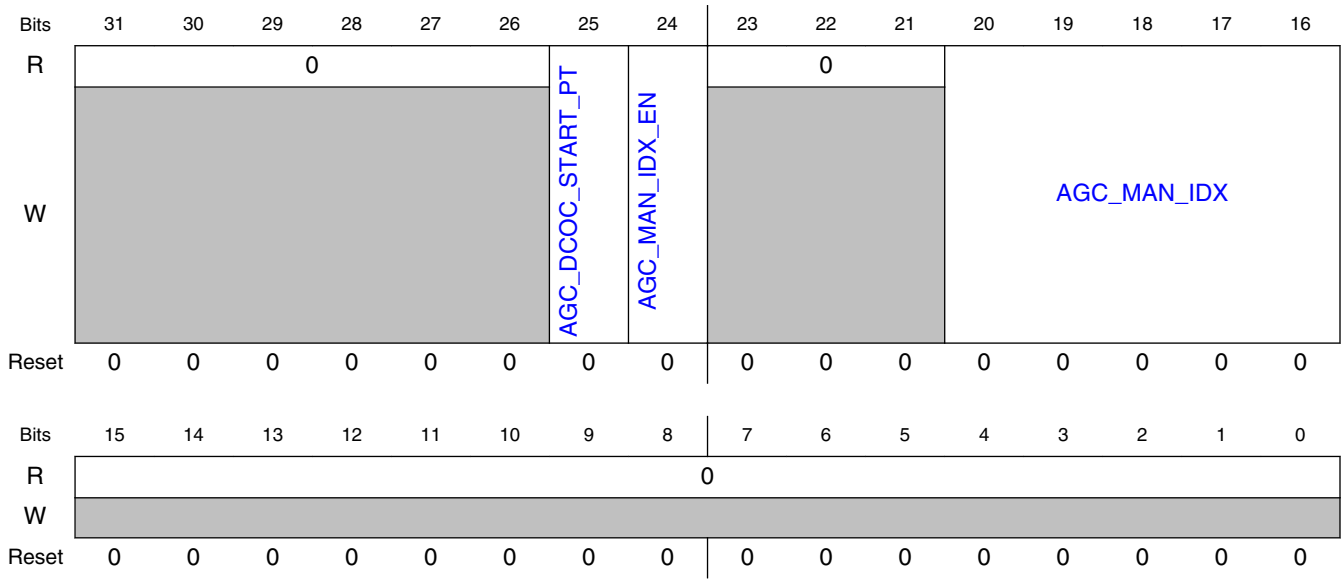
Field	Function
31-10 —	Reserved.
9-0 RX_CH_FILT_H11	RX Channel Filter Coefficient 11 RX Channel Filter Coefficient 11, 10-bit signed fractional

A.2.3.75 AGC Manual AGC Index (AGC_MAN_AGC_IDX)

A.2.3.75.1 Offset

Register	Offset
AGC_MAN_AGC_IDX	1D0h

A.2.3.75.2 Diagram



A.2.3.75.3 Fields

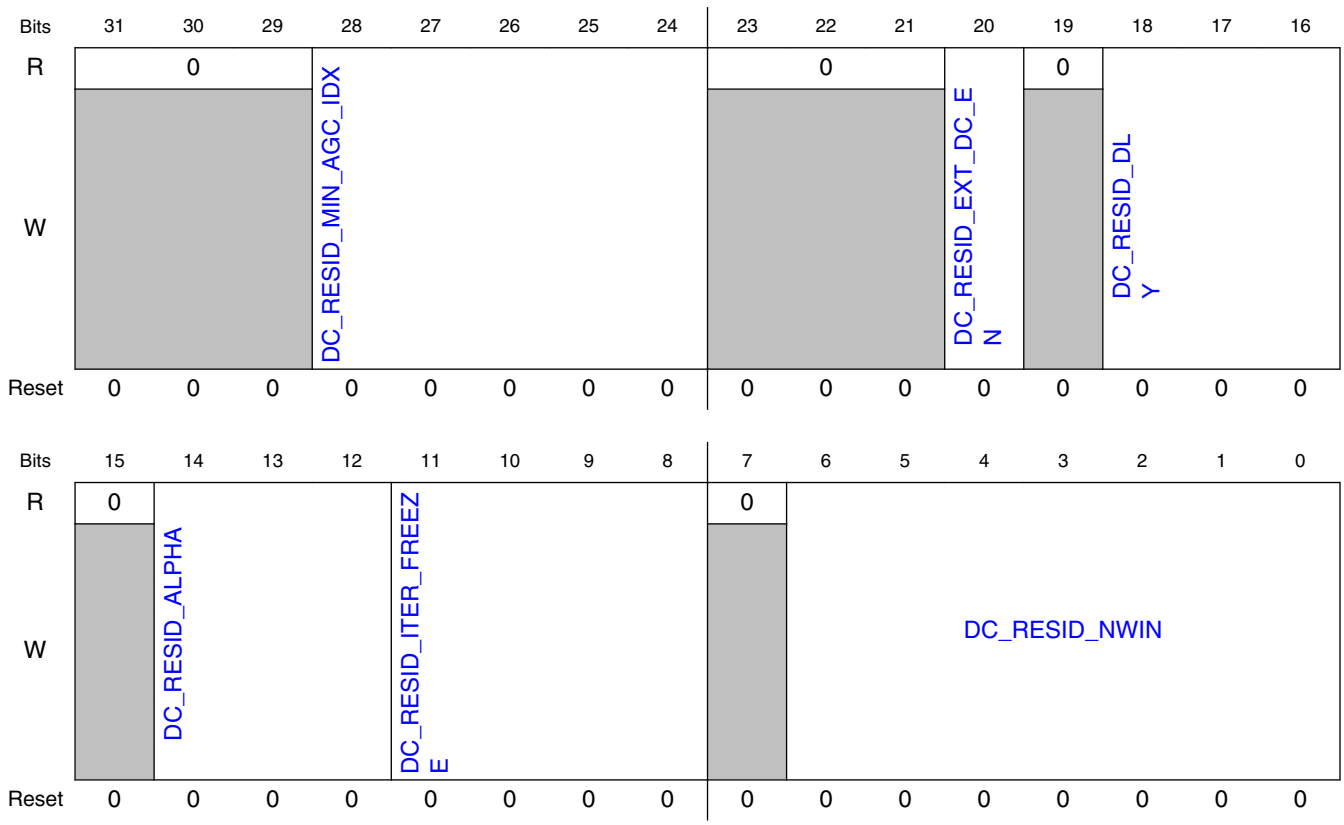
Field	Function
31-26 —	Reserved.
25 AGC_DCOC_START_PT	AGC DCOC Start Point When set, the starting value for the AGC gain table index after DCOC calibration will be AGC_MAN_IDX instead of index 26.
24 AGC_MAN_IDX_EN	AGC Manual Index Enable When set, the AGC gain table index is overridden using AGC_MAN_IDX.
23-21 —	Reserved.
20-16 AGC_MAN_IDX	AGC Manual Index AGC gain table index override value (when AGC_MAN_IDX_EN is set), or AGC gain table index starting point after DCOC calibration (when AGC_DCOC_START_PT is set).
15-0 —	Reserved.

A.2.3.76 DC Residual Control (DC_RESID_CTRL)

A.2.3.76.1 Offset

Register	Offset
DC_RESID_CTRL	1D4h

A.2.3.76.2 Diagram



A.2.3.76.3 Fields

Field	Function
31-29 —	Reserved.
28-24 DC_RESID_MIN_AGC_IDX	DC Residual Minimum AGC Table Index Specifies the minimum AGC table index value at which DC residual can be enabled. E.g., if this is 5'd0, then the DC residual is enabled for all AGC gain table indexes (assuming RX_DC_RESID_EN is set) if this is 5'd20, then tracking is enabled for AGC gain table indexes 20-26 (assuming RX_DC_RESID_EN is set).
23-21 —	Reserved.

Table continues on the next page...

Field	Function
20 DC_RESID_EXT_DC_EN	DC Residual External DC Enable 0b - External DC disable. The DC Residual activates at a delay specified by DC_RESID_DLY after an AGC gain change pulse. The DC Residual is initialized with a DC offset of 0. 1b - External DC enable. The DC residual activates after the DCOC's tracking hold timer expires. The DC Residual is initialized with the DC estimate from the DCOC tracking estimator.
19 —	Reserved.
18-16 DC_RESID_DLY	DC Residual Delay The delay from activation of the DC residual block to the time when it starts processing. For a 32MHz reference clock, the delay in microseconds matches the value programmed. For other clock frequencies, the delay is scaled based on the clock period. Supported values: 0-7. This delay is only used when EXT_DC_EN=0.
15 —	Reserved.
14-12 DC_RESID_ALPHA	DC Residual Alpha The Alpha parameter controls the rate at which the DC estimate is updated. The update factor is 2^(-Alpha).
11-8 DC_RESID_ITER_FREEZE	DC Residual Iteration Freeze Number of windows of DC_RESID_NWIN samples before the DC is frozen. Supported values: 1-8
7 —	Reserved.
6-0 DC_RESID_NWIN	DC Residual NWIN Number of samples in a window.

A.2.3.77 DC Residual Estimate (DC_RESID_EST)

A.2.3.77.1 Offset

Register	Offset
DC_RESID_EST	1D8h

A.2.3.77.2 Diagram

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

A.2.3.77.3 Fields

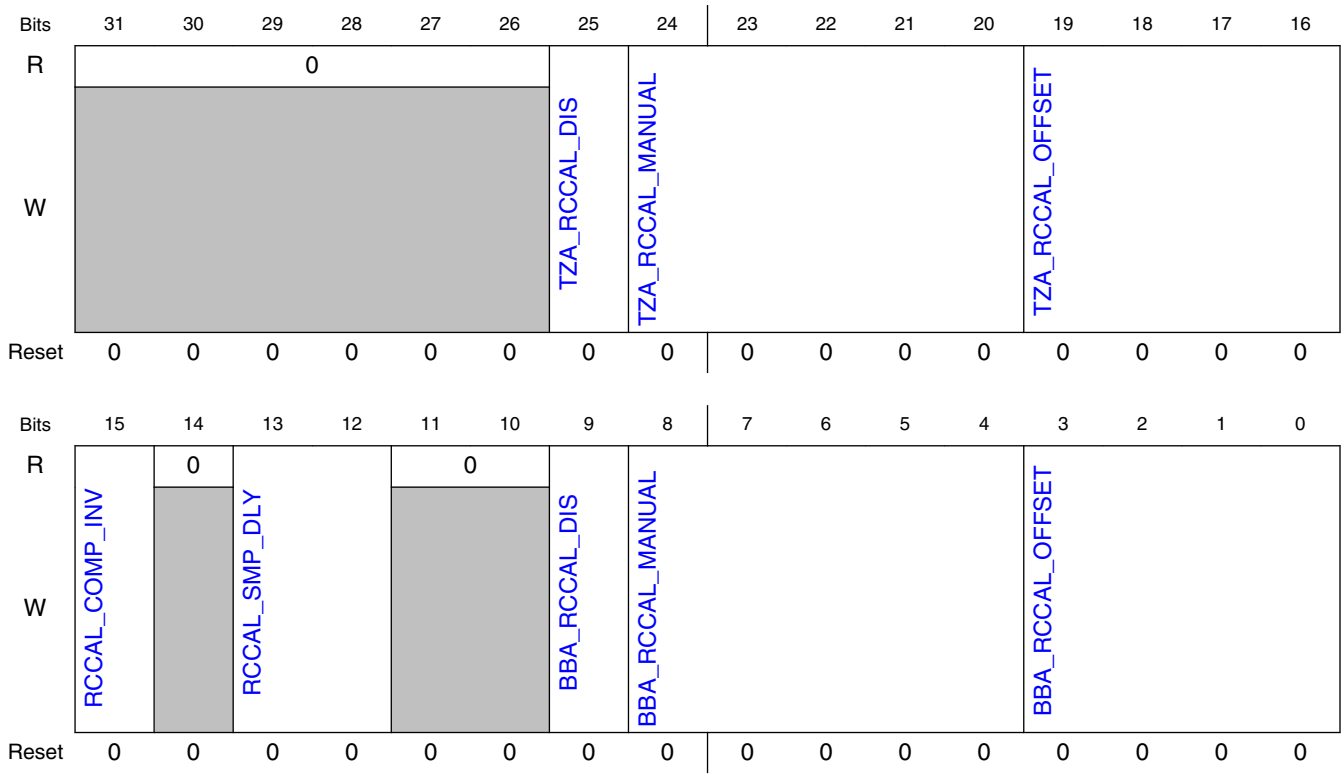
Field	Function
31-29 —	Reserved.
28-16 DC_RESID_OFFSET_Q	DC Residual Offset Q Reflects the current DC residual offset estimate for Q channel. Format is s11.1. This is provided for debug and characterization purposes only.
15-13 —	Reserved.
12-0 DC_RESID_OFFSET_I	DC Residual Offset I Reflects the current DC residual offset estimate for I channel. Format is s11.1. This is provided for debug and characterization purposes only.

A.2.3.78 RX RC Calibration Control0 (RX_RCCAL_CTRL0)

A.2.3.78.1 Offset

Register	Offset
RX_RCCAL_CTRL0	1DCh

A.2.3.78.2 Diagram



A.2.3.78.3 Fields

Field	Function
31-26 —	Reserved.
25 TZA_RCCAL_DIS	TZA RC Calibration Disable 0b - TZA RC Calibration is enabled 1b - TZA RC Calibration is disabled
24-20 TZA_RCCAL_MANUAL	TZA RC Calibration manual value If TZA_RCCAL_DIS bit is set, this value is used for the TZA calibration.
19-16 TZA_RCCAL_OFFSET	TZA RC Calibration value offset Offset added to the RC Calibration code to determine the calibration value used for the TZA. Format is s3.
15 RCCAL_COMP_INV	RC Calibration comp_out Invert 0b - The comp_out signal polarity is NOT inverted 1b - The comp_out signal polarity is inverted
14 —	Reserved.
13-12	RC Calibration Sample Delay

Table continues on the next page...

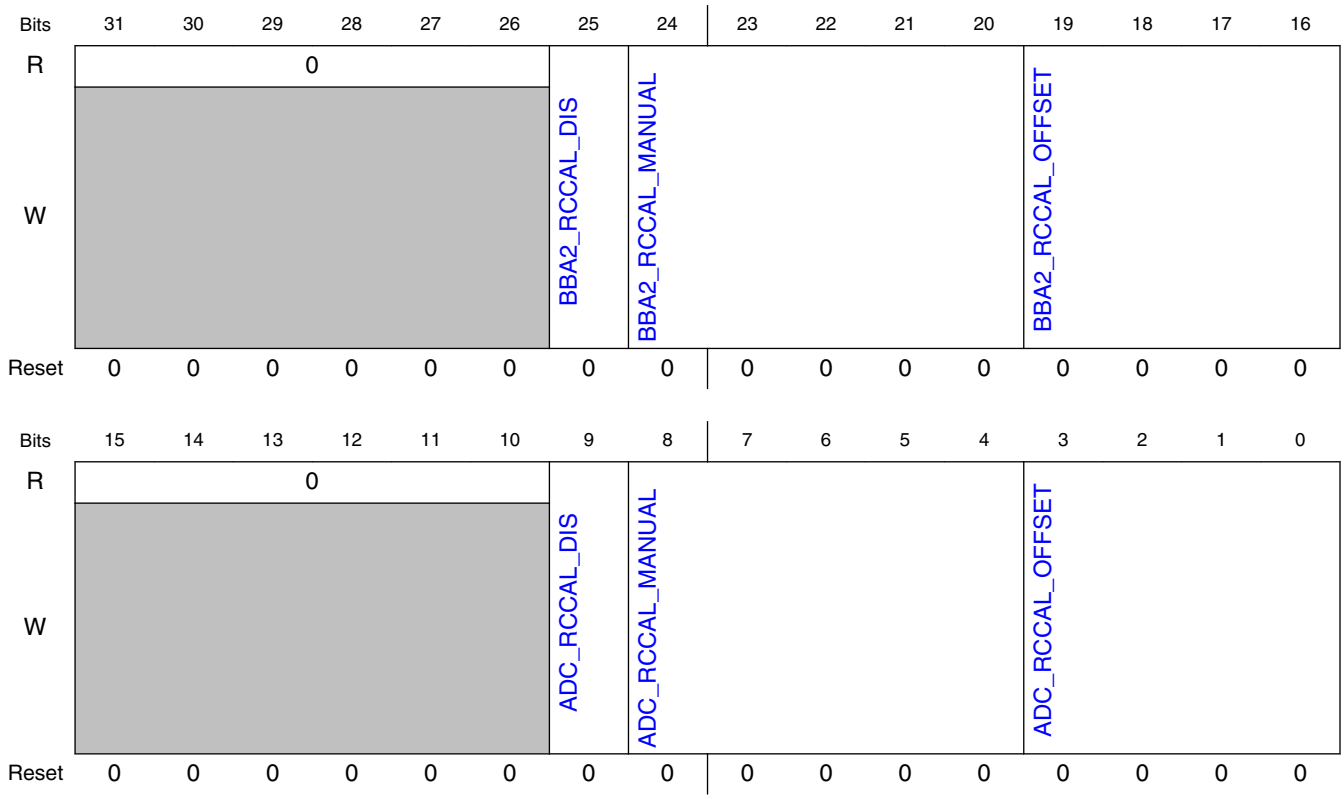
Field	Function
RCCAL_SMP_DLY	00b - The comp_out signal is sampled 0 clk cycle after sample signal is deasserted 01b - The comp_out signal is sampled 1 clk cycle after sample signal is deasserted 10b - The comp_out signal is sampled 2 clk cycle after sample signal is deasserted 11b - The comp_out signal is sampled 3 clk cycle after sample signal is deasserted
11-10 —	Reserved.
9 BBA_RCCAL_DIS	BBA RC Calibration Disable 0b - BBA RC Calibration is enabled 1b - BBA RC Calibration is disabled
8-4 BBA_RCCAL_MANUAL	BBA RC Calibration manual value If BBA_RCCAL_DIS bit is set, this value is used for the BBA calibration.
3-0 BBA_RCCAL_OFFSET	BBA RC Calibration value offset Offset added to the RC Calibration code to determine the calibration value used for the BBA. Format is s3.

A.2.3.79 RX RC Calibration Control1 (RX_RCCAL_CTRL1)

A.2.3.79.1 Offset

Register	Offset
RX_RCCAL_CTRL1	1E0h

A.2.3.79.2 Diagram



A.2.3.79.3 Fields

Field	Function
31-26 —	Reserved.
25 BBA2_RCCAL_DIS	BBA2 RC Calibration Disable 0b - BBA2 RC Calibration is enabled 1b - BBA2 RC Calibration is disabled
24-20 BBA2_RCCAL_MANUAL	BBA2 RC Calibration manual value If BBA2_RCCAL_DIS bit is set, this value is used for the BBA2 calibration.
19-16 BBA2_RCCAL_OFFSET	BBA2 RC Calibration value offset Offset added to the RC Calibration code to determine the calibration value used for the BBA2. Format is s3.
15-10 —	Reserved.
9 ADC_RCCAL_DIS	ADC RC Calibration Disable 0b - ADC RC Calibration is enabled 1b - ADC RC Calibration is disabled

Table continues on the next page...

Field	Function
8-4 ADC_RCCAL_MANUAL	ADC RC Calibration manual value If ADC_RCCAL_DIS bit is set, this value is used for the ADC calibration.
3-0 ADC_RCCAL_OFFSET	ADC RC Calibration value offset Offset added to the RC Calibration code to determine the calibration value used for the ADC. Format is s3.

A.2.3.80 RX RC Calibration Status (RX_RCCAL_STAT)

A.2.3.80.1 Offset

Register	Offset
RX_RCCAL_STAT	1E4h

A.2.3.80.2 Diagram

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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BBA2_RCCAL					ADC_RCCAL					RCCAL_CODE				
W																
Reset	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0

A.2.3.80.3 Fields

Field	Function
31-26 —	Reserved.
25-21 TZA_RCCAL	TZA RC Calibration The RC Calibration value used for TZA
20-16 BBA_RCCAL	BBA RC Calibration The RC Calibration value used for BBA
15	Reserved.

Table continues on the next page...

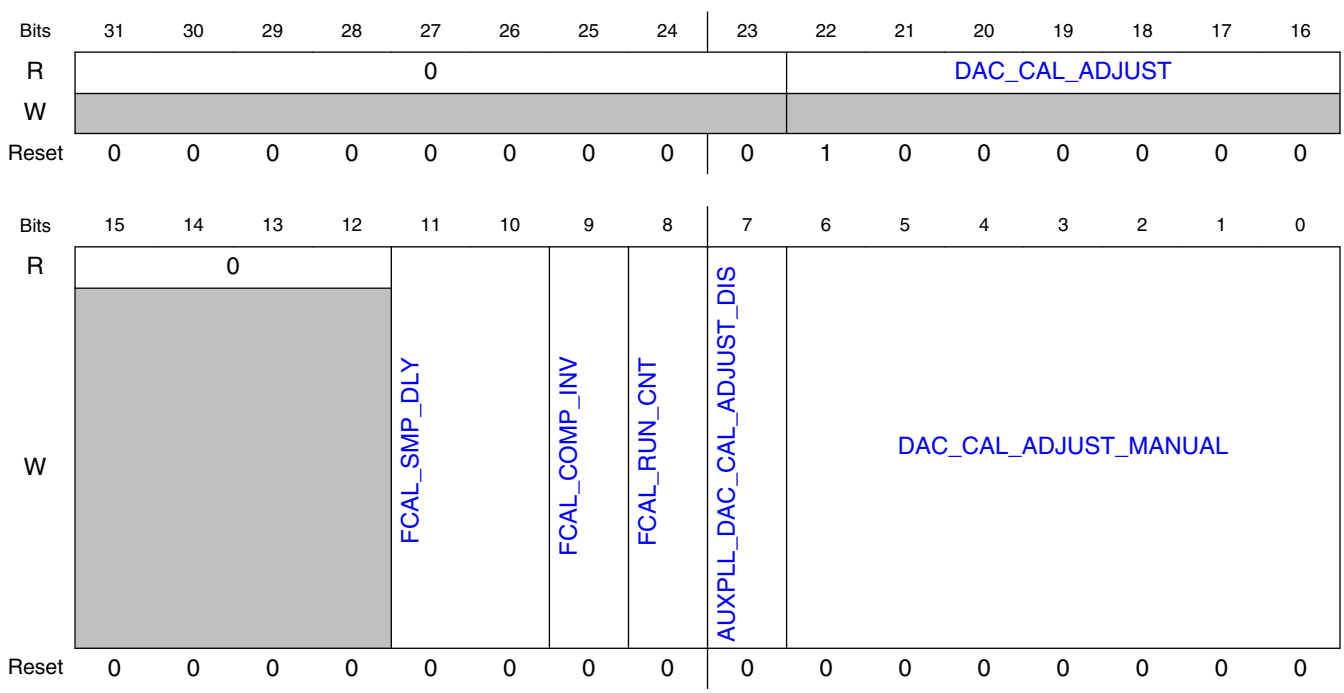
Field	Function
—	
14-10 BBA2_RCCAL	BBA2 RC Calibration The RC Calibration value used for BBA2
9-5 ADC_RCCAL	ADC RC Calibration The RC Calibration value used for ADC
4-0 RCCAL_CODE	RC Calibration code The RC Calibration code currently applied to the calibration circuit

A.2.3.81 Aux PLL Frequency Calibration Control (AUXPLL_FCAL_CTRL)

A.2.3.81.1 Offset

Register	Offset
AUXPLL_FCAL_CTRL	1E8h

A.2.3.81.2 Diagram



A.2.3.81.3 Fields

Field	Function
31-23 —	Reserved.
22-16 DAC_CAL_ADJUST	Aux PLL DAC Calibration Adjust value The DAC calibration adjust value applied to the calibration circuit.
15-12 —	Reserved.
11-10 FCAL_SMP_DLY	Aux PLL Frequency Calibration Sample Delay 00b - The count signal is sampled 1 clk cycle after fcal_run signal is deasserted 01b - The count signal is sampled 2 clk cycle after fcal_run signal is deasserted 10b - The count signal is sampled 3 clk cycle after fcal_run signal is deasserted 11b - The count signal is sampled 4 clk cycle after fcal_run signal is deasserted
9 FCAL_COMP_INV	Aux PLL Frequency Calibration Comparison Invert 0b - (Default) The comparison associated with the count is not inverted. 1b - The comparison associated with the count is inverted
8 FCAL_RUN_CNT	Aux PLL Frequency Calibration Run Count 0b - Run count is 256 clock cycles 1b - Run count is 512 clock cycles
7 AUXPLL_DAC_CAL_ADJUST_DIS	Aux PLL Frequency Calibration Disable 0b - Calibration is enabled 1b - Calibration is disabled
6-0 DAC_CAL_ADJUST_MANUAL	Aux PLL Frequency DAC Calibration Adjust Manual value If AUXPLL_DAC_CAL_ADJUST_DIS bit is set, this DAC calibration adjust value is applied to the calibration circuit.

A.2.3.82 Aux PLL Frequency Calibration Count 6 (AUXPLL_FCAL_CNT6)

A.2.3.82.1 Offset

Register	Offset
AUXPLL_FCAL_CNT6	1ECh

A.2.3.82.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						FCAL_BESTDIFF									
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	0						FCAL_COUNT_6									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.82.3 Fields

Field	Function
31-26 —	Reserved.
25-16 FCAL_BESTDIFF	Aux PLL Frequency Calibration Best Difference The smallest absolute difference between the count value output by the calibration circuit and the expected count value found during the calibration sequence.
15-10 —	Reserved.
9-0 FCAL_COUNT_6	Aux PLL Frequency Calibration Count 6 The count value output by the calibration circuit for calibration phase 6.

A.2.3.83 Aux PLL Frequency Calibration Count 5 and 4 (AUXPLL_FCAL_CNT5_4)

A.2.3.83.1 Offset

Register	Offset
AUXPLL_FCAL_CNT5_4	1F0h

A.2.3.83.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						FCAL_COUNT_5									
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	0						FCAL_COUNT_4									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.83.3 Fields

Field	Function
31-26 —	Reserved.
25-16 FCAL_COUNT_5	Aux PLL Frequency Calibration Count 5 The count value output by the calibration circuit for calibration phase 5.
15-10 —	Reserved.
9-0 FCAL_COUNT_4	Aux PLL Frequency Calibration Count 4 The count value output by the calibration circuit for calibration phase 4.

A.2.3.84 Aux PLL Frequency Calibration Count 3 and 2 (AUXPLL_FCAL_CNT3_2)

A.2.3.84.1 Offset

Register	Offset
AUXPLL_FCAL_CNT3_2	1F4h

A.2.3.84.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						FCAL_COUNT_3									
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	0						FCAL_COUNT_2									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.84.3 Fields

Field	Function
31-26 —	Reserved.
25-16 FCAL_COUNT_3	Aux PLL Frequency Calibration Count 3 The count value output by the calibration circuit for calibration phase 3.
15-10 —	Reserved.
9-0 FCAL_COUNT_2	Aux PLL Frequency Calibration Count 2 The count value output by the calibration circuit for calibration phase 2.

A.2.3.85 Aux PLL Frequency Calibration Count 1 and 0 (AUXPLL_FCAL_CNT1_0)

A.2.3.85.1 Offset

Register	Offset
AUXPLL_FCAL_CNT1_0	1F8h

A.2.3.85.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						FCAL_COUNT_1									
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	0						FCAL_COUNT_0									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.3.85.3 Fields

Field	Function
31-26 —	Reserved.
25-16 FCAL_COUNT_1	Frequency Calibration Count 1 The count value output by the calibration circuit for calibration phase 1.
15-10 —	Reserved.
9-0 FCAL_COUNT_0	Frequency Calibration Count 0 The count value output by the calibration circuit for calibration phase 0.

A.2.4 XCVR_TSM register descriptions

A.2.4.1 XCVR_TSM_ADDR Memory map

XCVR_TSM base address: 4005_C2C0h

Offset	Register	Width (In bits)	Access	Reset value
0h	TSM CONTROL (CTRL)	32	RW	FF00_4000h
4h	TSM END OF SEQUENCE (END_OF_SEQ)	32	RW	6766_6A63h
8h	PA POWER (PA_POWER)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
Ch	PA RAMP TABLE 0 (PA_RAMP_TBL0)	32	RW	0604_0201h
10h	PA RAMP TABLE 1 (PA_RAMP_TBL1)	32	RW	1410_0C08h
14h	PA RAMP TABLE 2 (PA_RAMP_TBL2)	32	RW	2822_1C18h
18h	PA RAMP TABLE 3 (PA_RAMP_TBL3)	32	RW	3C36_302Ch
24h	TSM RECYCLE COUNT (RECYCLE_COUNT)	32	RW	001A_0464h
28h	TSM FAST WARMUP CONTROL 1 (FAST_CTRL1)	32	RW	0000_FF00h
2Ch	TSM FAST WARMUP CONTROL 2 (FAST_CTRL2)	32	RW	FFFF_FFFFh
30h	TSM_TIMING00 (TIMING00)	32	RW	6700_6A00h
34h	TSM_TIMING01 (TIMING01)	32	RW	6700_6A00h
38h	TSM_TIMING02 (TIMING02)	32	RW	6700_FFFFh
3Ch	TSM_TIMING03 (TIMING03)	32	RW	6700_6A00h
40h	TSM_TIMING04 (TIMING04)	32	RW	6700_6A00h
44h	TSM_TIMING05 (TIMING05)	32	RW	6700_6A00h
48h	TSM_TIMING06 (TIMING06)	32	RW	6700_6A00h
4Ch	TSM_TIMING07 (TIMING07)	32	RW	0500_0500h
50h	TSM_TIMING08 (TIMING08)	32	RW	0300_0300h
54h	TSM_TIMING09 (TIMING09)	32	RW	0300_0300h
58h	TSM_TIMING10 (TIMING10)	32	RW	6703_6A03h
5Ch	TSM_TIMING11 (TIMING11)	32	RW	FFFF_6A03h
60h	TSM_TIMING12 (TIMING12)	32	RW	6703_FFFFh
64h	TSM_TIMING13 (TIMING13)	32	RW	1600_4A00h
68h	TSM_TIMING14 (TIMING14)	32	RW	672F_645Fh
6Ch	TSM_TIMING15 (TIMING15)	32	RW	6703_6A03h
70h	TSM_TIMING16 (TIMING16)	32	RW	671A_FFFFh
74h	TSM_TIMING17 (TIMING17)	32	RW	FFFF_6A5Ah
78h	TSM_TIMING18 (TIMING18)	32	RW	6705_6A05h
7Ch	TSM_TIMING19 (TIMING19)	32	RW	1605_4A05h
80h	TSM_TIMING20 (TIMING20)	32	RW	6705_6A05h
84h	TSM_TIMING21 (TIMING21)	32	RW	6704_6A04h
88h	TSM_TIMING22 (TIMING22)	32	RW	6704_FFFFh
8Ch	TSM_TIMING23 (TIMING23)	32	RW	FFFF_6A04h
90h	TSM_TIMING24 (TIMING24)	32	RW	1600_4A00h
94h	TSM_TIMING25 (TIMING25)	32	RW	671B_FFFFh
98h	TSM_TIMING26 (TIMING26)	32	RW	FFFF_6A5Ah
9Ch	TSM_TIMING27 (TIMING27)	32	RW	671E_FFFFh
A0h	TSM_TIMING28 (TIMING28)	32	RW	1F1E_FFFFh
A4h	TSM_TIMING29 (TIMING29)	32	RW	671C_FFFFh
A8h	TSM_TIMING30 (TIMING30)	32	RW	671E_FFFFh
ACh	TSM_TIMING31 (TIMING31)	32	RW	671D_FFFFh
B0h	TSM_TIMING32 (TIMING32)	32	RW	671B_FFFFh

Table continues on the next page...

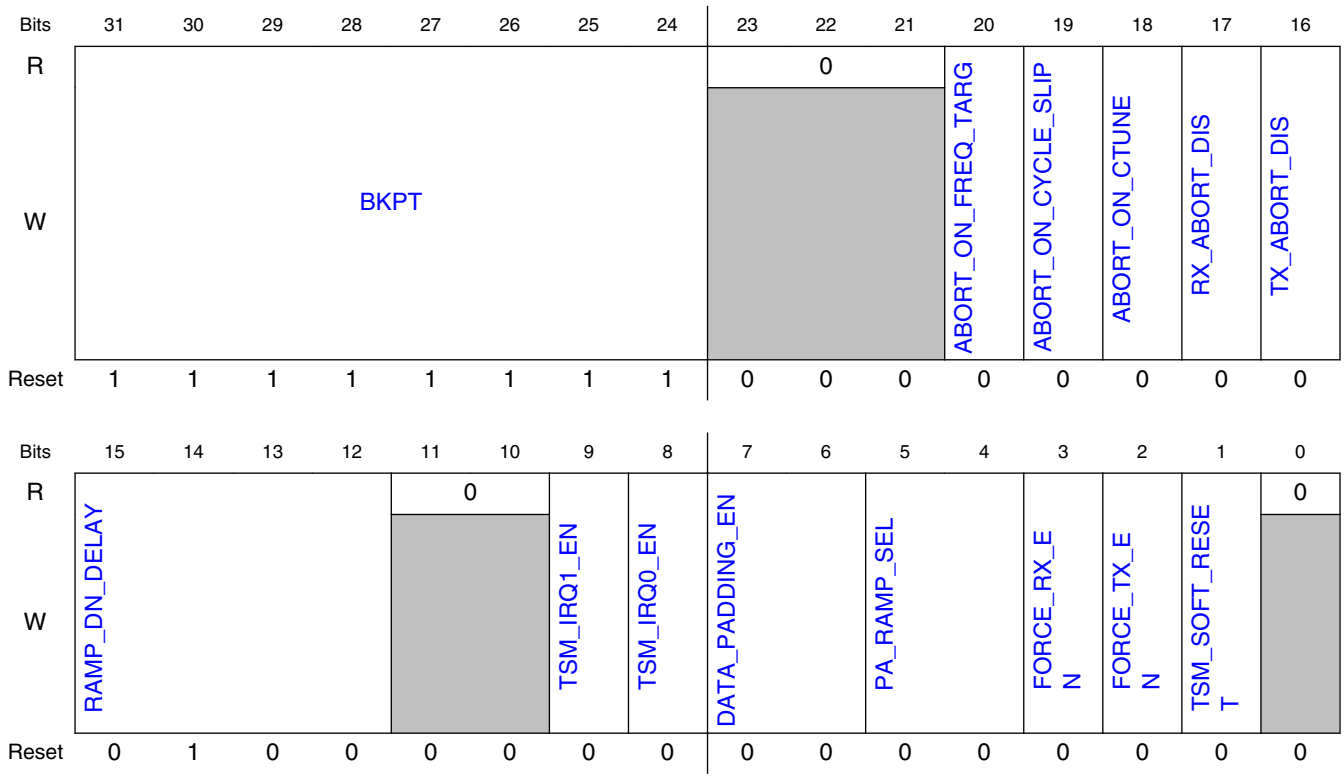
Offset	Register	Width (In bits)	Access	Reset value
B4h	TSM_TIMING33 (TIMING33)	32	RW	671E_FFFFh
B8h	TSM_TIMING34 (TIMING34)	32	RW	6705_6A05h
BCh	TSM_TIMING35 (TIMING35)	32	RW	FFFF_6A5Dh
C0h	TSM_TIMING36 (TIMING36)	32	RW	6764_FFFFh
C4h	TSM_TIMING37 (TIMING37)	32	RW	6564_FFFFh
C8h	TSM_TIMING38 (TIMING38)	32	RW	670C_6A40h
CCh	TSM_TIMING39 (TIMING39)	32	RW	6764_FFFFh
D0h	TSM_TIMING40 (TIMING40)	32	RW	6724_FFFFh
D4h	TSM_TIMING41 (TIMING41)	32	RW	2524_FFFFh
D8h	TSM_TIMING42 (TIMING42)	32	RW	FFFF_FFFFh
DCh	TSM_TIMING43 (TIMING43)	32	RW	FFFF_FFFFh
E0h	TSM_TIMING44 (TIMING44)	32	RW	FFFF_FFFFh
E4h	TSM_TIMING45 (TIMING45)	32	RW	FFFF_FFFFh
E8h	TSM_TIMING46 (TIMING46)	32	RW	FFFF_FFFFh
ECh	TSM_TIMING47 (TIMING47)	32	RW	FFFF_FFFFh
F0h	TSM_TIMING48 (TIMING48)	32	RW	FFFF_FFFFh
F4h	TSM_TIMING49 (TIMING49)	32	RW	FFFF_FFFFh
F8h	TSM_TIMING50 (TIMING50)	32	RW	FFFF_FFFFh
FCh	TSM_TIMING51 (TIMING51)	32	RW	6703_FFFFh
100h	TSM_TIMING52 (TIMING52)	32	RW	1504_FFFFh
104h	TSM_TIMING53 (TIMING53)	32	RW	6704_FFFFh
108h	TSM_TIMING54 (TIMING54)	32	RW	1504_FFFFh
10Ch	TSM_TIMING55 (TIMING55)	32	RW	671E_FFFFh
110h	TSM_TIMING56 (TIMING56)	32	RW	671E_FFFFh
114h	TSM_TIMING57 (TIMING57)	32	RW	1A03_FFFFh
118h	TSM_TIMING58 (TIMING58)	32	RW	FFFF_6A03h
11Ch	TSM OVERRIDE REGISTER 0 (OVRD0)	32	RW	0000_0000h
120h	TSM OVERRIDE REGISTER 1 (OVRD1)	32	RW	0000_0000h
124h	TSM OVERRIDE REGISTER 2 (OVRD2)	32	RW	0000_0000h
128h	TSM OVERRIDE REGISTER 3 (OVRD3)	32	RW	0000_0000h

A.2.4.2 TSM CONTROL (CTRL)

A.2.4.2.1 Offset

Register	Offset
CTRL	0h

A.2.4.2.2 Diagram



A.2.4.2.3 Fields

Field	Function
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
20 ABORT_ON_FREQ_TARG	Abort On Frequency Target Lock Detect Failure 0b - don't allow TSM abort on Frequency Target Unlock Detect 1b - allow TSM abort on Frequency Target Unlock Detect
19 ABORT_ON_CYCLE_SLIP	Abort On Cycle Slip Lock Detect Failure 0b - don't allow TSM abort on Cycle Slip Unlock Detect 1b - allow TSM abort on Cycle Slip Unlock Detect
18	Abort On Coarse Tune Lock Detect Failure 0b - don't allow TSM abort on Coarse Tune Unlock Detect

Table continues on the next page...

Field	Function
ABORT_ON_CTUNE	1b - allow TSM abort on Coarse Tune Unlock Detect
17 RX_ABORT_DIS	Receive Abort Disable RX Abort disable. When set, prevents PLL unlock events during RX sequences from aborting the sequence.
16 TX_ABORT_DIS	Transmit Abort Disable TX Abort disable. When set, prevents PLL unlock events during TX sequences from aborting the sequence.
15-12 RAMP_DN_DELAY	PA Ramp Down Delay Delays the start of the PA Ramp Down, relative to the start of the TSM warmdown, by *N* microseconds, where *N*=RAMP_DN_DELAY. Range is 0 to 15us.
11-10 —	Reserved
9 TSM_IRQ1_EN	TSM_IRQ1 Enable/Disable bit TSM_IRQ1 Enable/Disable bit. Intended for debug purposes only. 0b - TSM_IRQ1 is disabled 1b - TSM_IRQ1 is enabled
8 TSM_IRQ0_EN	TSM_IRQ0 Enable/Disable bit TSM_IRQ0 Enable/Disable bit. Intended for debug purposes only. 0b - TSM_IRQ0 is disabled 1b - TSM_IRQ0 is enabled
7-6 DATA_PADDING_EN	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]. 00b - Disable TX Data Padding 01b - Enable TX Data Padding
5-4 PA_RAMP_SEL	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 Ramp Table. There are always 16 ramp steps, if ramping is enabled. The following table lists the total ramp duration for each PA_RAMP_SEL setting: 0: No ramp 1: 1.0µs 2: 2.0µs 3: 4.0µs The following table lists the duration of each ramp step for each PA_RAMP_SEL setting: 0: No ramp 1: 2 Reference Clocks (0.0625µs) 2: 4 Reference Clocks (0.125µs) 3: 8 Reference Clocks (0.25µs)
3 FORCE_RX_EN	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. 0b - TSM Idle 1b - TSM executes a RX sequence

Table continues on the next page...

Field	Function
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. 0b - TSM Idle 1b - TSM executes a TX sequence
1 TSM_SOFT_RESET	TSM Soft Reset Writing 1 to this bit returns TSM to its IDLE state, with TSM_COUNT=0. Writing 0 to this bit removes the forced-reset condition and resumes normal operation. The bit is not self-clearing. For contingency purposes only 0b - TSM Soft Reset removed. Normal operation. 1b - TSM Soft Reset engaged. TSM forced to IDLE, and holds there until the bit is cleared.
0 —	Reserved

A.2.4.3 TSM END OF SEQUENCE (END_OF_SEQ)

A.2.4.3.1 Offset

Register	Offset
END_OF_SEQ	4h

A.2.4.3.2 Diagram

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

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

A.2.4.3.3 Fields

Field	Function
31-24	End of RX Warmdown

Table continues on the next page...

Field	Function
END_OF_RX_WD	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.
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.
7-0 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.

A.2.4.4 PA POWER (PA_POWER)

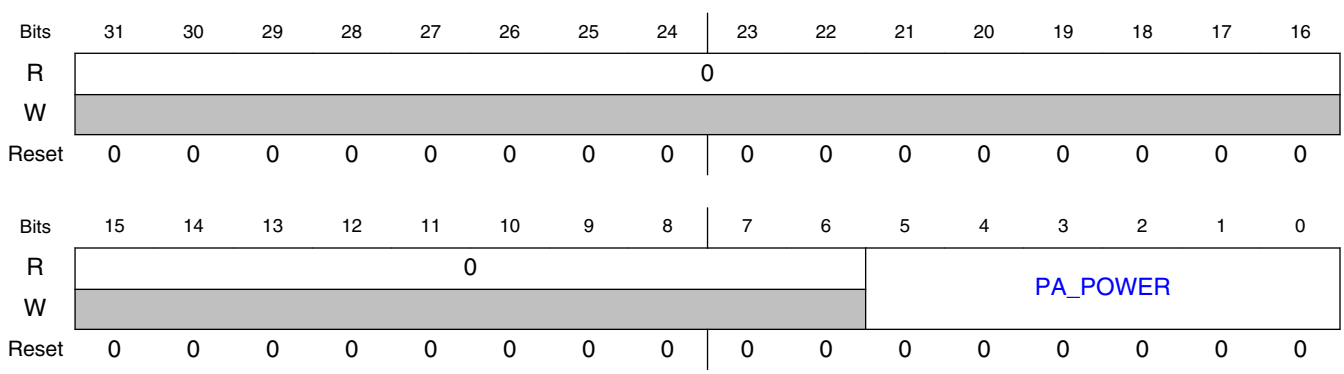
A.2.4.4.1 Offset

Register	Offset
PA_POWER	8h

A.2.4.4.2 Function

This contents of this register are used as PA target power when XCVR_CTRL[TGT_PWR_SRC] = 0

A.2.4.4.3 Diagram



A.2.4.4.4 Fields

Field	Function
31-6 —	Reserved
5-0 PA_POWER	PA POWER This contents of this register are used as PA target power when XCVR_CTRL[TGT_PWR_SRC] = 0.The valid values for this field are 0,1,2,4,6,8,…,62 (in increasing order of PA power). That is, above 1, only even numbers are permitted in the PA_POWER bitfield. Odd values of 3..63 are not permitted and will result in unexpected behavior.

A.2.4.5 PA RAMP TABLE 0 (PA_RAMP_TBL0)

A.2.4.5.1 Offset

Register	Offset
PA_RAMP_TBL0	Ch

A.2.4.5.2 Function

PA Ramp Table 0

A.2.4.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		PA_RAMP3						0		PA_RAMP2					
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PA_RAMP1						0		PA_RAMP0					
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

A.2.4.5.4 Fields

Field	Function
31-30 —	Reserved
29-24 PA_RAMP3	PA_RAMP3 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 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_RAMP3 cannot exceed target power (enforced by PA ramping logic).
23-22 —	Reserved
21-16 PA_RAMP2	PA_RAMP2 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 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_RAMP2 cannot exceed target power (enforced by PA ramping logic).
15-14 —	Reserved
13-8 PA_RAMP1	PA_RAMP1 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_RAMP1 cannot exceed target power (enforced by PA ramping logic).
7-6 —	Reserved
5-0 PA_RAMP0	PA_RAMP0 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_pa_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_RAMP0 cannot exceed target power (enforced by PA ramping logic). When PA ramping is enabled, the contents of PA_RAMP0 are also presented to the PA during sequence-idle conditions.

A.2.4.6 PA RAMP TABLE 1 (PA_RAMP_TBL1)

A.2.4.6.1 Offset

Register	Offset
PA_RAMP_TBL1	10h

A.2.4.6.2 Function

PA Ramp Table 1

A.2.4.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		PA_RAMP7						0		PA_RAMP6					
W																
Reset	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PA_RAMP5						0		PA_RAMP4					
W																
Reset	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0

A.2.4.6.4 Fields

Field	Function
31-30 —	Reserved
29-24 PA_RAMP7	PA_RAMP7 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 eighth ramp step. During PA ramp down, the contents of this register are the PA power value during the eighth-to-last ramp step. In both cases, PA_RAMP7 cannot exceed target power (enforced by PA ramping logic).
23-22 —	Reserved
21-16 PA_RAMP6	PA_RAMP6 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 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_RAMP6 cannot exceed target power (enforced by PA ramping logic).
15-14 —	Reserved
13-8 PA_RAMP5	PA_RAMP5 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 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_RAMP5 cannot exceed target power (enforced by PA ramping logic).
7-6 —	Reserved
5-0 PA_RAMP4	PA_RAMP4

Field	Function
	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 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_RAMP4 cannot exceed target power (enforced by PA ramping logic).

A.2.4.7 PA RAMP TABLE 2 (PA_RAMP_TBL2)

A.2.4.7.1 Offset

Register	Offset
PA_RAMP_TBL2	14h

A.2.4.7.2 Function

PA Ramp Table 2

A.2.4.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		PA_RAMP11						0		PA_RAMP10					
W																
Reset	0	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PA_RAMP9						0		PA_RAMP8					
W																
Reset	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0

A.2.4.7.4 Fields

Field	Function
31-30 —	Reserved
29-24 PA_RAMP11	PA_RAMP11 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 twelvth ramp step. During PA ramp down, the

Table continues on the next page...

Field	Function
	contents of this register are the PA power value during the twelvth-to-last ramp step. In both cases, PA_RAMP11 cannot exceed target power (enforced by PA ramping logic).
23-22 —	Reserved
21-16 PA_RAMP10	PA_RAMP10 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 eleventh ramp step. During PA ramp down, the contents of this register are the PA power value during the eleventh-to-last ramp step. In both cases, PA_RAMP10 cannot exceed target power (enforced by PA ramping logic).
15-14 —	Reserved
13-8 PA_RAMP9	PA_RAMP9 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 tenth ramp step. During PA ramp down, the contents of this register are the PA power value during the tenth-to-last ramp step. In both cases, PA_RAMP9 cannot exceed target power (enforced by PA ramping logic).
7-6 —	Reserved
5-0 PA_RAMP8	PA_RAMP8 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 ninth ramp step. During PA ramp down, the contents of this register are the PA power value during the ninth-to-last ramp step. In both cases, PA_RAMP8 cannot exceed target power (enforced by PA ramping logic).

A.2.4.8 PA RAMP TABLE 3 (PA_RAMP_TBL3)

A.2.4.8.1 Offset

Register	Offset
PA_RAMP_TBL3	18h

A.2.4.8.2 Function

PA Ramp Table 3

A.2.4.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		PA_RAMP15						0		PA_RAMP14					
W																
Reset	0	0	1	1	1	1	0	0	0	0	1	1	0	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PA_RAMP13						0		PA_RAMP12					
W																
Reset	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0	0

A.2.4.8.4 Fields

Field	Function
31-30 —	Reserved
29-24 PA_RAMP15	PA_RAMP15 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 sixteenth (final) ramp step. During PA ramp down, the contents of this register are the PA power value during the sixteenth-to-last (first) ramp step. In both cases, PA_RAMP15 cannot exceed target power (enforced by PA ramping logic).
23-22 —	Reserved
21-16 PA_RAMP14	PA_RAMP14 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 fifteenth ramp step. During PA ramp down, the contents of this register are the PA power value during the fifteenth-to-last ramp step. In both cases, PA_RAMP14 cannot exceed target power (enforced by PA ramping logic).
15-14 —	Reserved
13-8 PA_RAMP13	PA_RAMP13 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 fourteenth ramp step. During PA ramp down, the contents of this register are the PA power value during the fourteenth-to-last ramp step. In both cases, PA_RAMP13 cannot exceed target power (enforced by PA ramping logic).
7-6 —	Reserved
5-0 PA_RAMP12	PA_RAMP12 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 thirteenth ramp step. During PA ramp down, the contents of this register are the PA power value during the thirteenth-to-last ramp step. In both cases, PA_RAMP12 cannot exceed target power (enforced by PA ramping logic).

A.2.4.9 TSM RECYCLE COUNT (RECYCLE_COUNT)

A.2.4.9.1 Offset

Register	Offset
RECYCLE_COUNT	24h

A.2.4.9.2 Function

This register contains the TSM Recycle "Jump-to" points for the 3 types of TSM Recycle

A.2.4.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								RECYCLE_COUNT2							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RECYCLE_COUNT1								RECYCLE_COUNT0							
W																
Reset	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0

A.2.4.9.4 Fields

Field	Function
31-24 —	Reserved
23-16 RECYCLE_COUNT2	TSM RX Recycle Count 2 The RECYCLE_COUNT2[7:0] register determines the TSM count value to which the TSM "recycles" when, in LPPS mode, an LPPS power save cycle completes and the receiver is switched back on, necessitating a foreshortened RX warmup. The intention is for this register to be programmed to a TSM count value that is just prior to the assertion of rx_lna_en, but there are no restrictions on programming this register. An LPPS recycle is a command from the LPPS state machine to TSM, requesting a jump from the TSM ON phase back to a programmable point in the WARMUP phase, and resume counting from there.
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 802.15.4 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

Table continues on the next page...

Field	Function
	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, requesting a 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.
7-0 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 802.15.4 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. This register also determines the TSM count value to which the TSM recycles when theGeneric_FSK Link Layer asserts a tsm_recycle to TSM. 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 any protocol engine to TSM, requesting a 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".

A.2.4.10 TSM FAST WARMUP CONTROL 1 (FAST_CTRL1)

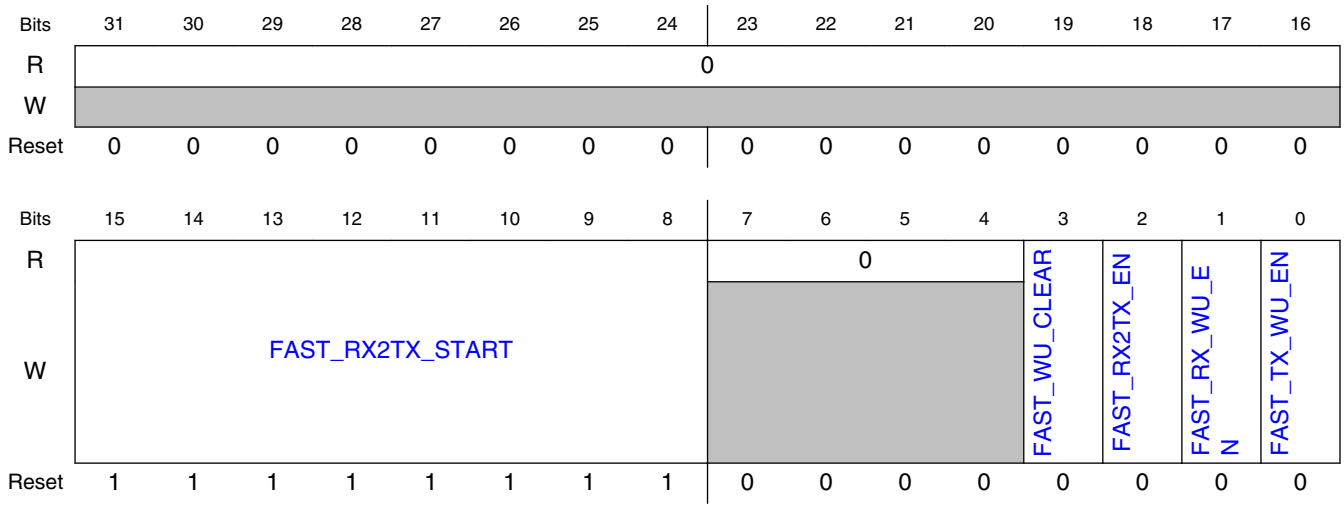
A.2.4.10.1 Offset

Register	Offset
FAST_CTRL1	28h

A.2.4.10.2 Function

This register provides enabling and control for Fast TSM Mode

A.2.4.10.3 Diagram



A.2.4.10.4 Fields

Field	Function
31-16 —	Reserved
15-8 FAST_RX2TX_START	These bits currently have no functionality.
7-4 —	Reserved
3 FAST_WU_CLEAR	Fast TSM Warmup Clear State This bit clears the RF channel memory in the PLL Digital Block, forcing the next TSM TX Warmup to be normal (not fast), and the next TSM RX Warmup to be normal (not fast), regardless of whether the RF channel has actually changed or not. This bit is not self-clearing. Write '1' to clear channel memory, then write '0' to proceed with TSM operations.
2 FAST_RX2TX_EN	Fast TSM RX-to-TX Transition Enable This bit currently has no functionality.
1 FAST_RX_WU_EN	Fast TSM RX Warmup Enable 0b - Fast TSM RX Warmups are disabled 1b - Fast TSM RX Warmups are enabled, if the RF channel has not changed since the last RX warmup, and for BLE mode, the RF channel is not an advertising channel.
0 FAST_TX_WU_EN	Fast TSM TX Warmup Enable 0b - Fast TSM TX Warmups are disabled 1b - Fast TSM TX Warmups are enabled, if the RF channel has not changed since the last TX warmup, and for BLE mode, the RF channel is not an advertising channel.

A.2.4.11 TSM FAST WARMUP CONTROL 2 (FAST_CTRL2)

A.2.4.11.1 Offset

Register	Offset
FAST_CTRL2	2Ch

A.2.4.11.2 Function

This register provides configuration for Fast TSM Mode

A.2.4.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FAST_DEST_RX								FAST_START_RX							
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	FAST_DEST_TX								FAST_START_TX							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

A.2.4.11.4 Fields

Field	Function
31-24 FAST_DEST_RX	Fast TSM RX "Jump-to" Point During a Fast RX Warmup, this value represents the ending TSM count for the portion of the warmup that is to be skipped over. If Fast RX Warmups are enabled, FAST_START_RX[7:0] will never be reached and will instead be replaced by the contents of this register, thereby executing the jump. Example: FAST_START_RX = 10 FAST_DEST_RX = 15 The TSM will count as follows: … 7,8,9,15,16,17 …
23-16 FAST_START_RX	Fast TSM RX "Jump-from" Point During a Fast RX Warmup, this value represents the starting TSM count for the portion of the warmup that is to be skipped over. If Fast RX Warmups are enabled, this count will never be reached; its place in the TSM warmup will be replaced by the contents of FAST_DEST_RX[7:0], thereby executing the jump.
15-8	Fast TSM TX "Jump-to" Point

Table continues on the next page...

Field	Function
FAST_DEST_TX	During a Fast TX Warmup, this value represents the ending TSM count for the portion of the warmup that is to be skipped over. If Fast TX Warmups are enabled, FAST_START_TX[7:0] will never be reached and will instead be replaced by the contents of this register, thereby executing the jump. Example: FAST_START_TX = 10 FAST_DEST_TX = 15 The TSM will count as follows: … 7,8,9,15,16,17 …
7-0 FAST_START_TX	Fast TSM TX "Jump-from" Point During a Fast TX Warmup, this value represents the starting TSM count for the portion of the warmup that is to be skipped over. If Fast TX Warmups are enabled, this count will never be reached; its place in the TSM warmup will be replaced by the contents of FAST_DEST_TX[7:0], thereby executing the jump.

A.2.4.12 TSM_TIMING00 (TIMING00)

A.2.4.12.1 Offset

Register	Offset
TIMING00	30h

A.2.4.12.2 Function

This register contains the timing values to control the assertion and deassertion times for both TX and RX sequences for the BB_LDO_HF_EN TSM signal or signal group.

A.2.4.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_HF_EN_RX_LO								BB_LDO_HF_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_HF_EN_TX_LO								BB_LDO_HF_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

A.2.4.12.4 Fields

Field	Function
31-24 BB_LDO_HF_EN_RX_LO	De-assertion time setting for BB_LDO_HF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_HF_EN signal or group will transition from HI to LO.
23-16 BB_LDO_HF_EN_RX_HI	Assertion time setting for BB_LDO_HF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_HF_EN signal or group will transition from LO to HI.
15-8 BB_LDO_HF_EN_TX_LO	De-assertion time setting for BB_LDO_HF_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_HF_EN signal or group will transition from HI to LO.
7-0 BB_LDO_HF_EN_TX_HI	Assertion time setting for BB_LDO_HF_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_HF_EN signal or group will transition from LO to HI.

A.2.4.13 TSM_TIMING01 (TIMING01)

A.2.4.13.1 Offset

Register	Offset
TIMING01	34h

A.2.4.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_ADCDAC_EN_RX_LO								BB_LDO_ADCDAC_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_ADCDAC_EN_TX_LO								BB_LDO_ADCDAC_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

A.2.4.13.3 Fields

Field	Function
31-24 BB_LDO_ADCDAC_EN_RX_LO	De-assertion time setting for BB_LDO_ADCDAC_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_ADCDAC_EN signal or group will transition from HI to LO.
23-16 BB_LDO_ADCDAC_EN_RX_HI	Assertion time setting for BB_LDO_ADCDAC_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_ADCDAC_EN signal or group will transition from LO to HI.
15-8 BB_LDO_ADCDAC_EN_TX_LO	De-assertion time setting for BB_LDO_ADCDAC_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_ADCDAC_EN signal or group will transition from HI to LO.
7-0 BB_LDO_ADCDAC_EN_TX_HI	Assertion time setting for BB_LDO_ADCDAC_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_ADCDAC_EN signal or group will transition from LO to HI.

A.2.4.14 TSM_TIMING02 (TIMING02)

A.2.4.14.1 Offset

Register	Offset
TIMING02	38h

A.2.4.14.2 Diagram

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

A.2.4.14.3 Fields

Field	Function
31-24 BB_LDO_BBA_EN_RX_LO	De-assertion time setting for BB_LDO_BBA_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_BBA_EN signal or group will transition from HI to LO.
23-16 BB_LDO_BBA_EN_RX_HI	Assertion time setting for BB_LDO_BBA_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_BBA_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.15 TSM_TIMING03 (TIMING03)

A.2.4.15.1 Offset

Register	Offset
TIMING03	3Ch

A.2.4.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_PD_EN_RX_LO								BB_LDO_PD_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_PD_EN_TX_LO								BB_LDO_PD_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

A.2.4.15.3 Fields

Field	Function
31-24 BB_LDO_PD_EN_RX_LO	De-assertion time setting for BB_LDO_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_PD_EN signal or group will transition from HI to LO.

Table continues on the next page...

Field	Function
23-16 BB_LDO_PD_EN_RX_HI	Assertion time setting for BB_LDO_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_PD_EN signal or group will transition from LO to HI.
15-8 BB_LDO_PD_EN_TX_LO	De-assertion time setting for BB_LDO_PD_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_PD_EN signal or group will transition from HI to LO.
7-0 BB_LDO_PD_EN_TX_HI	Assertion time setting for BB_LDO_PD_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_PD_EN signal or group will transition from LO to HI.

A.2.4.16 TSM_TIMING04 (TIMING04)

A.2.4.16.1 Offset

Register	Offset
TIMING04	40h

A.2.4.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_FDBK_EN_RX_LO								BB_LDO_FDBK_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_FDBK_EN_TX_LO								BB_LDO_FDBK_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

A.2.4.16.3 Fields

Field	Function
31-24 BB_LDO_FDBK_EN_RX_LO	De-assertion time setting for BB_LDO_FDBK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_EN signal or group will transition from HI to LO.
23-16	Assertion time setting for BB_LDO_FDBK_EN (RX)

Table continues on the next page...

Field	Function
BB_LDO_FDBK_EN_RX_HI	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for BB_LDO_FDBK_EN (TX)
BB_LDO_FDBK_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_EN signal or group will transition from HI to LO.
7-0	Assertion time setting for BB_LDO_FDBK_EN (TX)
BB_LDO_FDBK_EN_TX_HI	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_EN signal or group will transition from LO to HI.

A.2.4.17 TSM_TIMING05 (TIMING05)

A.2.4.17.1 Offset

Register	Offset
TIMING05	44h

A.2.4.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_VCOLO_EN_RX_LO								BB_LDO_VCOLO_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_VCOLO_EN_TX_LO								BB_LDO_VCOLO_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

A.2.4.17.3 Fields

Field	Function
31-24	De-assertion time setting for BB_LDO_VCOLO_EN (RX)
BB_LDO_VCOLO_EN_RX_LO	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_EN signal or group will transition from HI to LO.
23-16	Assertion time setting for BB_LDO_VCOLO_EN (RX)
BB_LDO_VCOLO_EN_RX_HI	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_EN signal or group will transition from LO to HI.

Table continues on the next page...

Field	Function
15-8 BB_LDO_VCOLO_EN_TX_LO	De-assertion time setting for BB_LDO_VCOLO_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_EN signal or group will transition from HI to LO.
7-0 BB_LDO_VCOLO_EN_TX_HI	Assertion time setting for BB_LDO_VCOLO_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_EN signal or group will transition from LO to HI.

A.2.4.18 TSM_TIMING06 (TIMING06)

A.2.4.18.1 Offset

Register	Offset
TIMING06	48h

A.2.4.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BB_LDO_VTREF_EN_RX_LO								BB_LDO_VTREF_EN_RX_HI							
Reset	0	1	1	0	0	1	1	1	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	BB_LDO_VTREF_EN_TX_LO								BB_LDO_VTREF_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

A.2.4.18.3 Fields

Field	Function
31-24 BB_LDO_VTREF_EN_RX_LO	De-assertion time setting for BB_LDO_VTREF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VTREF_EN signal or group will transition from HI to LO.
23-16 BB_LDO_VTREF_EN_RX_HI	Assertion time setting for BB_LDO_VTREF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VTREF_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for BB_LDO_VTREF_EN (TX)

Table continues on the next page...

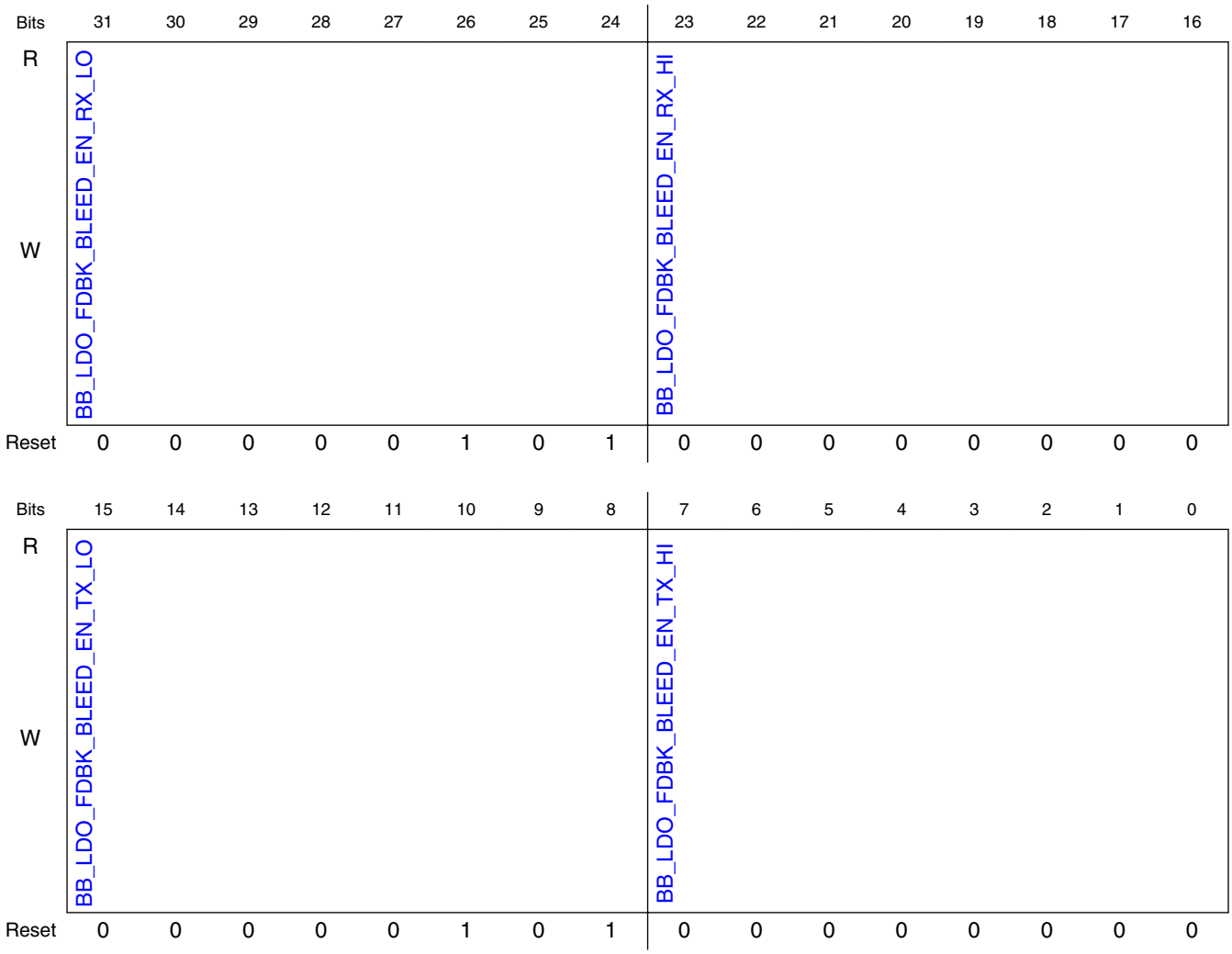
Field	Function
BB_LDO_VTREF_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VTREF_EN signal or group will transition from HI to LO.
7-0 BB_LDO_VTREF_EN_TX_HI	Assertion time setting for BB_LDO_VTREF_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VTREF_EN signal or group will transition from LO to HI.

A.2.4.19 TSM_TIMING07 (TIMING07)

A.2.4.19.1 Offset

Register	Offset
TIMING07	4Ch

A.2.4.19.2 Diagram



A.2.4.19.3 Fields

Field	Function
31-24 BB_LDO_FDBK_BLEED_EN_RX_LO	De-assertion time setting for BB_LDO_FDBK_BLEED_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_BLEED_EN signal or group will transition from HI to LO.
23-16 BB_LDO_FDBK_BLEED_EN_RX_HI	Assertion time setting for BB_LDO_FDBK_BLEED_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_BLEED_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for BB_LDO_FDBK_BLEED_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_BLEED_EN signal or group will transition from HI to LO.

Table continues on the next page...

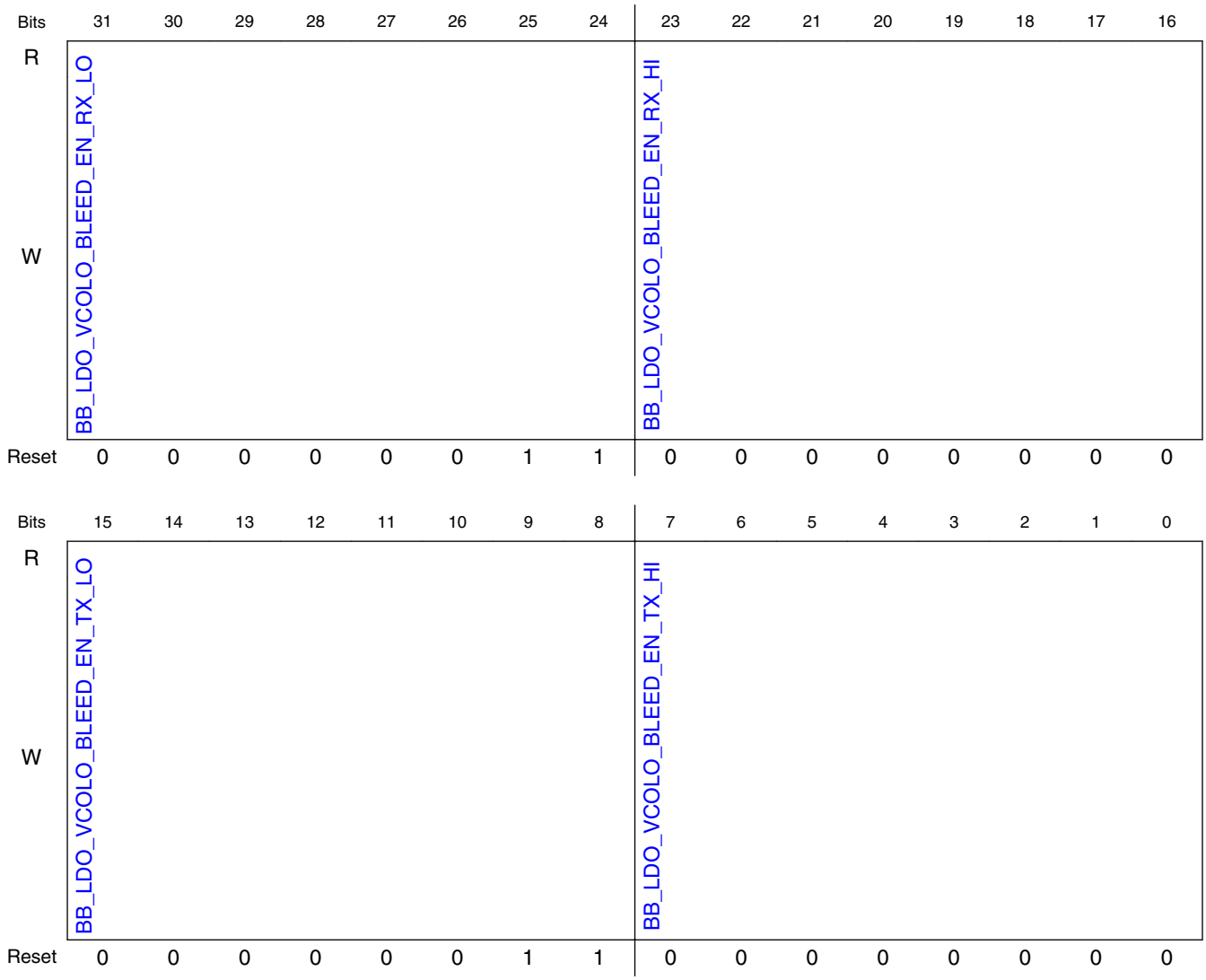
Field	Function
BB_LDO_FDBK_BLEED_EN_TX_LO	
7-0 BB_LDO_FDBK_BLEED_EN_TX_HI	Assertion time setting for BB_LDO_FDBK_BLEED_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_FDBK_BLEED_EN signal or group will transition from LO to HI.

A.2.4.20 TSM_TIMING08 (TIMING08)

A.2.4.20.1 Offset

Register	Offset
TIMING08	50h

A.2.4.20.2 Diagram



A.2.4.20.3 Fields

Field	Function
31-24 BB_LDO_VCOLO_BLEED_EN_RX_LO	De-assertion time setting for BB_LDO_VCOLO_BLEED_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_BLEED_EN signal or group will transition from HI to LO.
23-16 BB_LDO_VCOLO_BLEED_EN_RX_HI	Assertion time setting for BB_LDO_VCOLO_BLEED_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_BLEED_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for BB_LDO_VCOLO_BLEED_EN (TX)

Table continues on the next page...

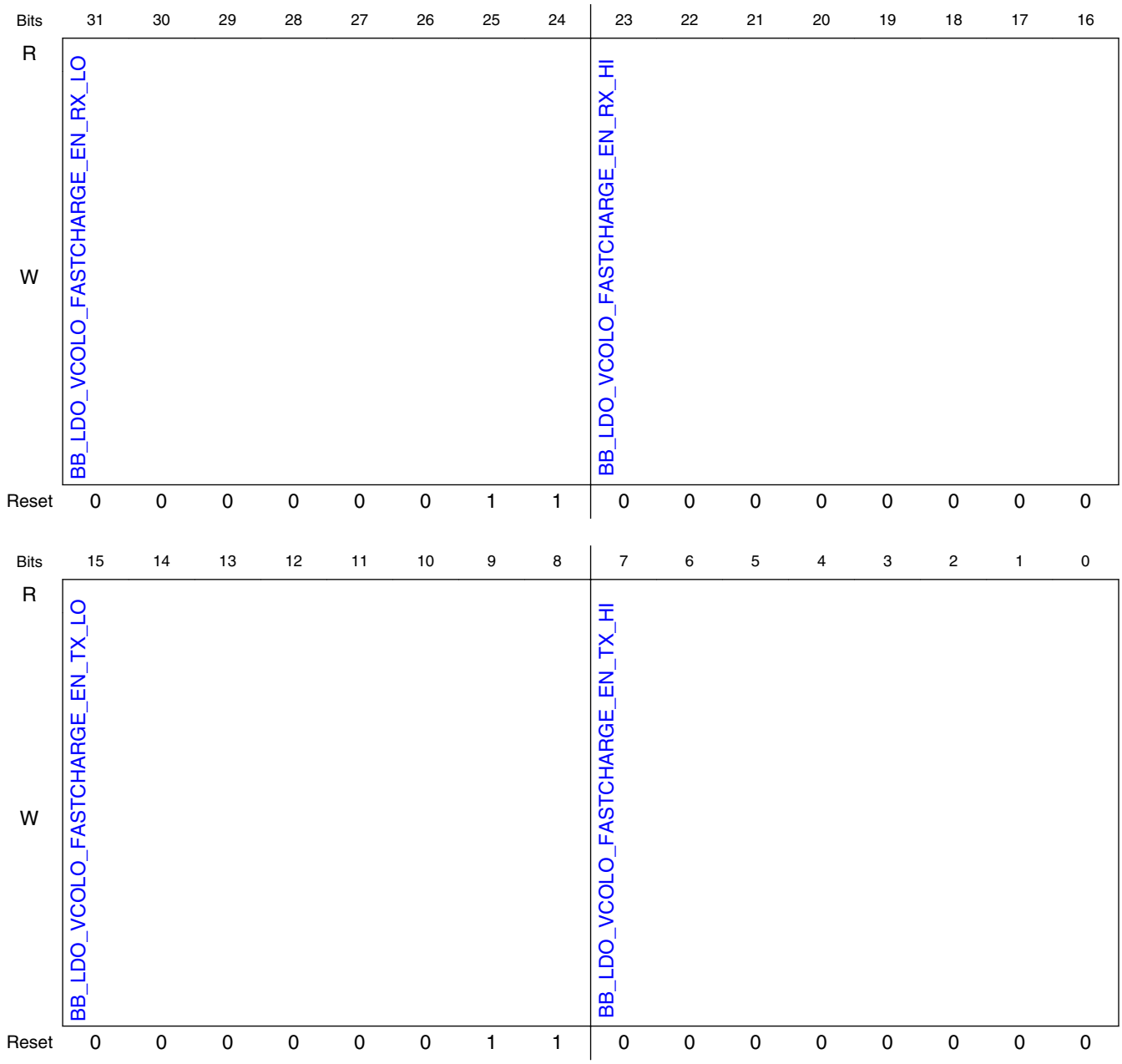
Field	Function
BB_LDO_VCOLO_BLEED_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_BLEED_EN signal or group will transition from HI to LO.
7-0 BB_LDO_VCOLO_BLEED_EN_TX_HI	Assertion time setting for BB_LDO_VCOLO_BLEED_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_BLEED_EN signal or group will transition from LO to HI.

A.2.4.21 TSM_TIMING09 (TIMING09)

A.2.4.21.1 Offset

Register	Offset
TIMING09	54h

A.2.4.21.2 Diagram



A.2.4.21.3 Fields

Field	Function
31-24 BB_LDO_VCOLO_FASTCHARGE_EN_RX_LO	De-assertion time setting for BB_LDO_VCOLO_FASTCHARGE_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_FASTCHARGE_EN signal or group will transition from HI to LO.
23-16	Assertion time setting for BB_LDO_VCOLO_FASTCHARGE_EN (RX)

Table continues on the next page...

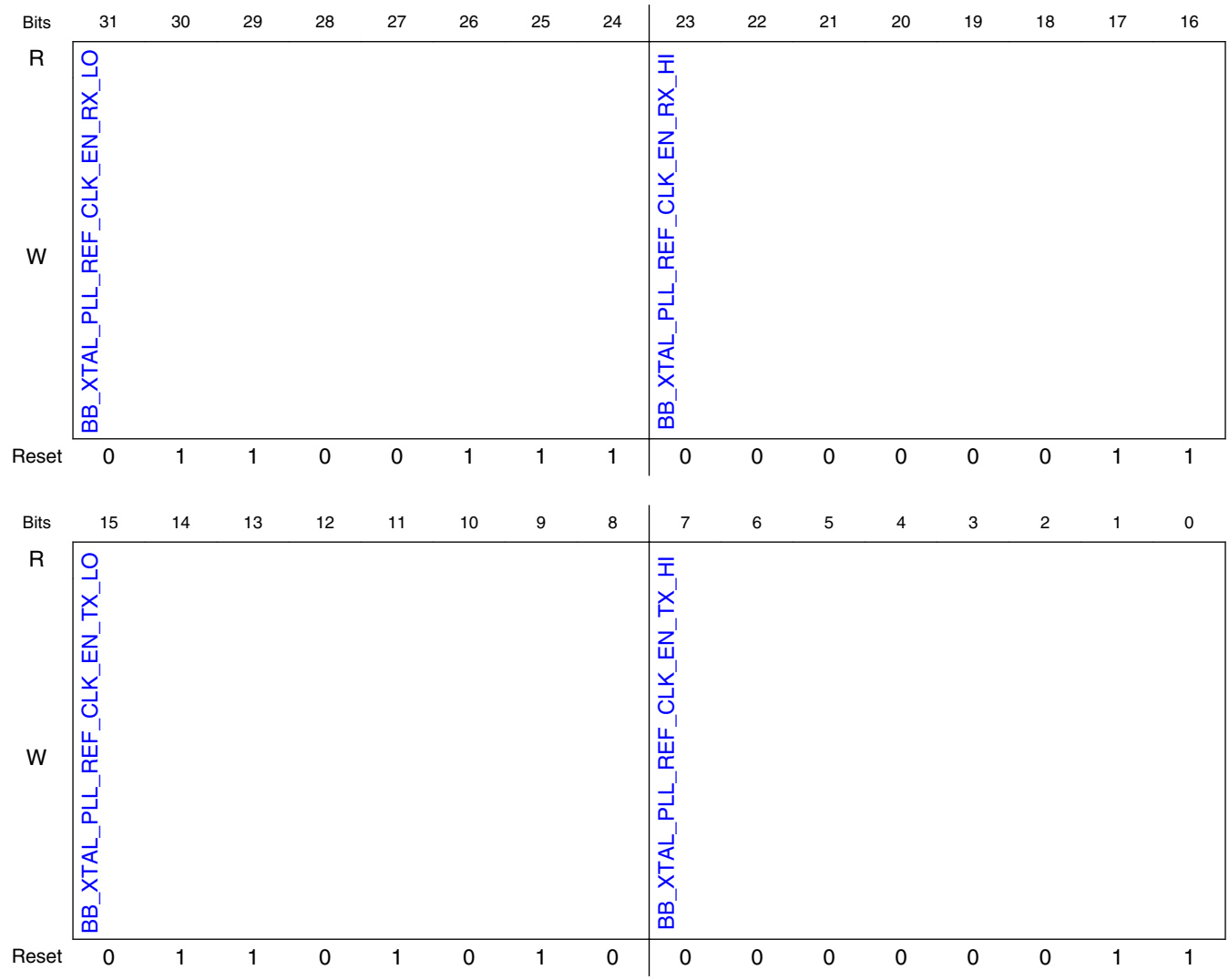
Field	Function
BB_LDO_VCOLO_FASTCHARGE_EN_RX_HI	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_FASTCHARGE_EN signal or group will transition from LO to HI.
15-8 BB_LDO_VCOLO_FASTCHARGE_EN_TX_LO	De-assertion time setting for BB_LDO_VCOLO_FASTCHARGE_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_FASTCHARGE_EN signal or group will transition from HI to LO.
7-0 BB_LDO_VCOLO_FASTCHARGE_EN_TX_HI	Assertion time setting for BB_LDO_VCOLO_FASTCHARGE_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_LDO_VCOLO_FASTCHARGE_EN signal or group will transition from LO to HI.

A.2.4.22 TSM_TIMING10 (TIMING10)

A.2.4.22.1 Offset

Register	Offset
TIMING10	58h

A.2.4.22.2 Diagram



A.2.4.22.3 Fields

Field	Function
31-24 BB_XTAL_PLL_REF_CLK_EN_RX_LO	De-assertion time setting for BB_XTAL_PLL_REF_CLK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_XTAL_PLL_REF_CLK_EN signal or group will transition from HI to LO.
23-16 BB_XTAL_PLL_REF_CLK_EN_RX_HI	Assertion time setting for BB_XTAL_PLL_REF_CLK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the BB_XTAL_PLL_REF_CLK_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for BB_XTAL_PLL_REF_CLK_EN (TX)

Table continues on the next page...

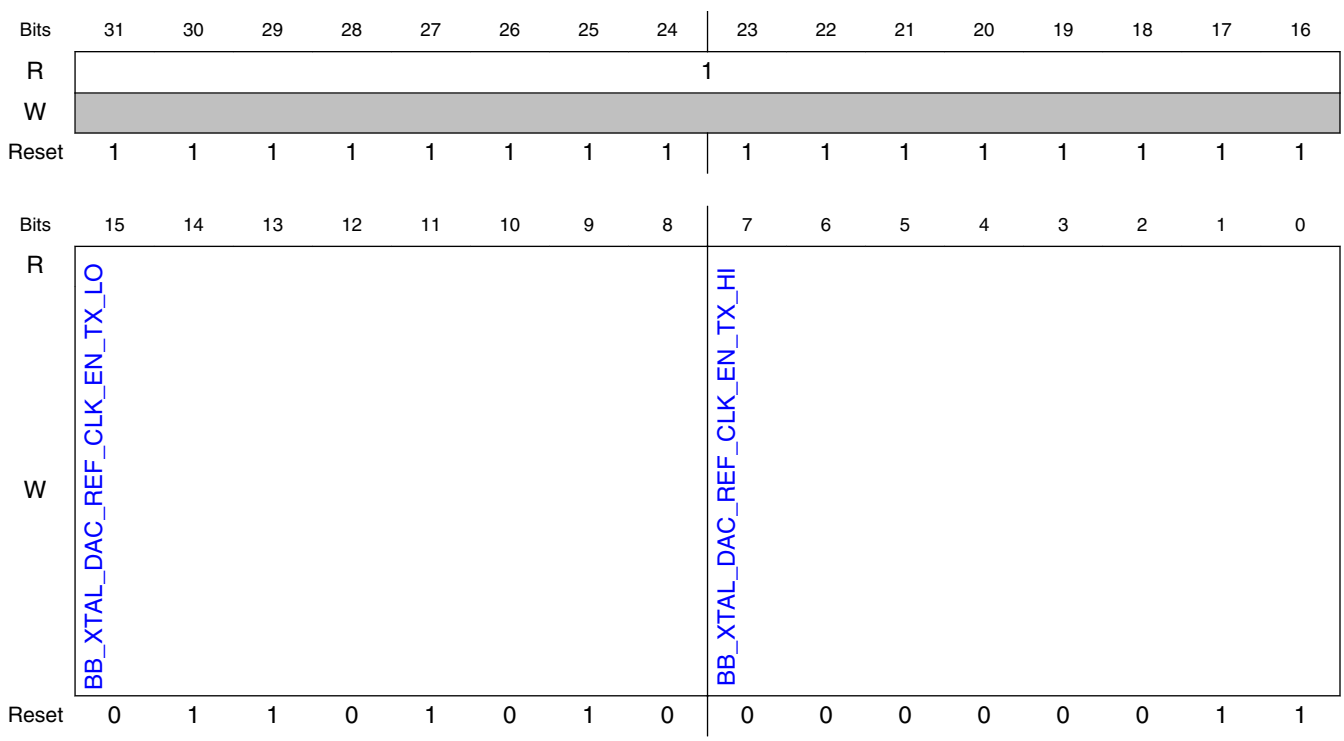
Field	Function
BB_XTAL_PLL_REF_CLK_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_XTAL_PLL_REF_CLK_EN signal or group will transition from HI to LO.
7-0 BB_XTAL_PLL_REF_CLK_EN_TX_HI	Assertion time setting for BB_XTAL_PLL_REF_CLK_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_XTAL_PLL_REF_CLK_EN signal or group will transition from LO to HI.

A.2.4.23 TSM_TIMING11 (TIMING11)

A.2.4.23.1 Offset

Register	Offset
TIMING11	5Ch

A.2.4.23.2 Diagram



A.2.4.23.3 Fields

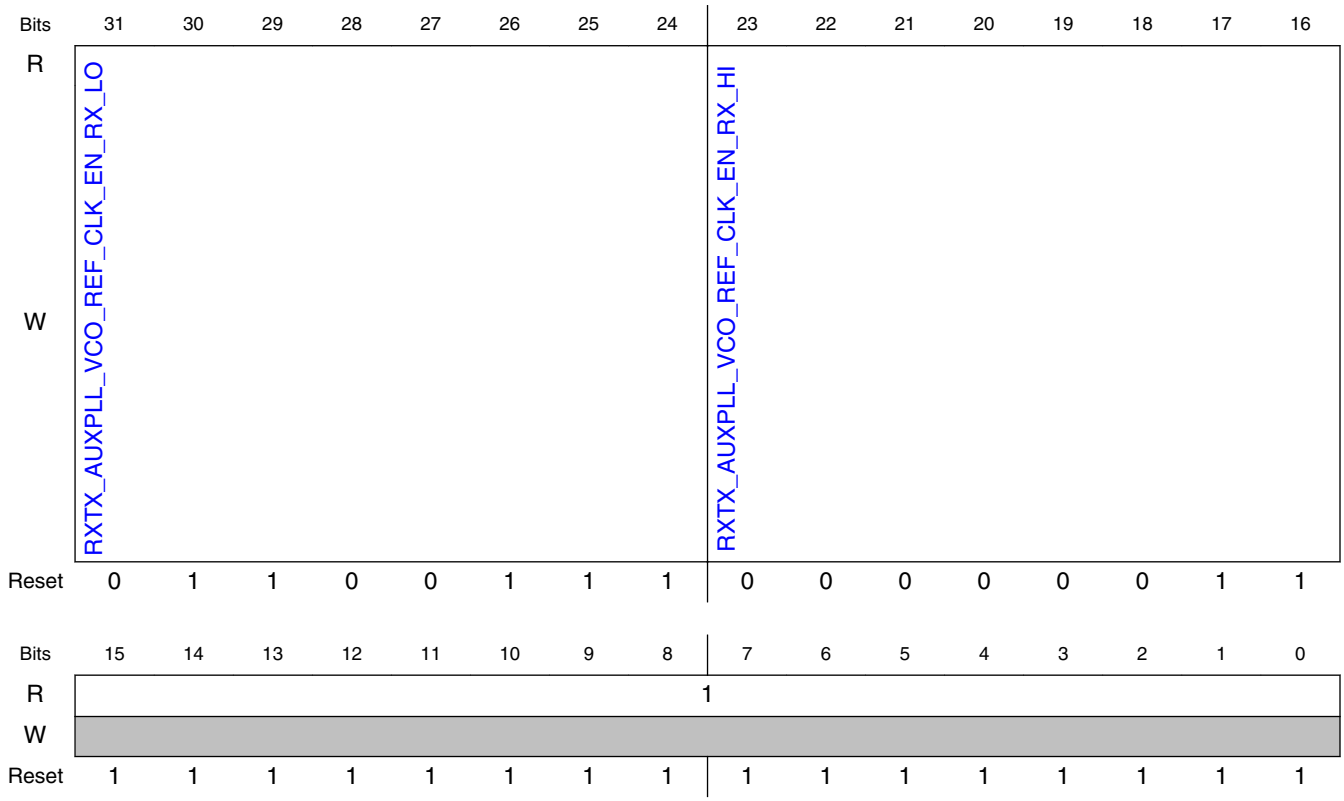
Field	Function
31-16 —	Reserved
15-8 BB_XTAL_DAC_REF_CLK_EN_TX_LO	De-assertion time setting for BB_XTAL_DAC_REF_CLK_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_XTAL_DAC_REF_CLK_EN signal or group will transition from HI to LO.
7-0 BB_XTAL_DAC_REF_CLK_EN_TX_HI	Assertion time setting for BB_XTAL_DAC_REF_CLK_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the BB_XTAL_DAC_REF_CLK_EN signal or group will transition from LO to HI.

A.2.4.24 TSM_TIMING12 (TIMING12)

A.2.4.24.1 Offset

Register	Offset
TIMING12	60h

A.2.4.24.2 Diagram



A.2.4.24.3 Fields

Field	Function
31-24 RXTX_AUXPLL_VCO_REF_CLK_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_VCO_REF_CLK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_VCO_REF_CLK_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_VCO_REF_CLK_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_VCO_REF_CLK_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_VCO_REF_CLK_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.25 TSM_TIMING13 (TIMING13)

A.2.4.25.1 Offset

Register	Offset
TIMING13	64h

A.2.4.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PLL_LOOP_IS_OPEN_RX_LO								PLL_LOOP_IS_OPEN_RX_HI							
Reset	0	0	0	1	0	1	1	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	PLL_LOOP_IS_OPEN_TX_LO								PLL_LOOP_IS_OPEN_TX_HI							
Reset	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

A.2.4.25.3 Fields

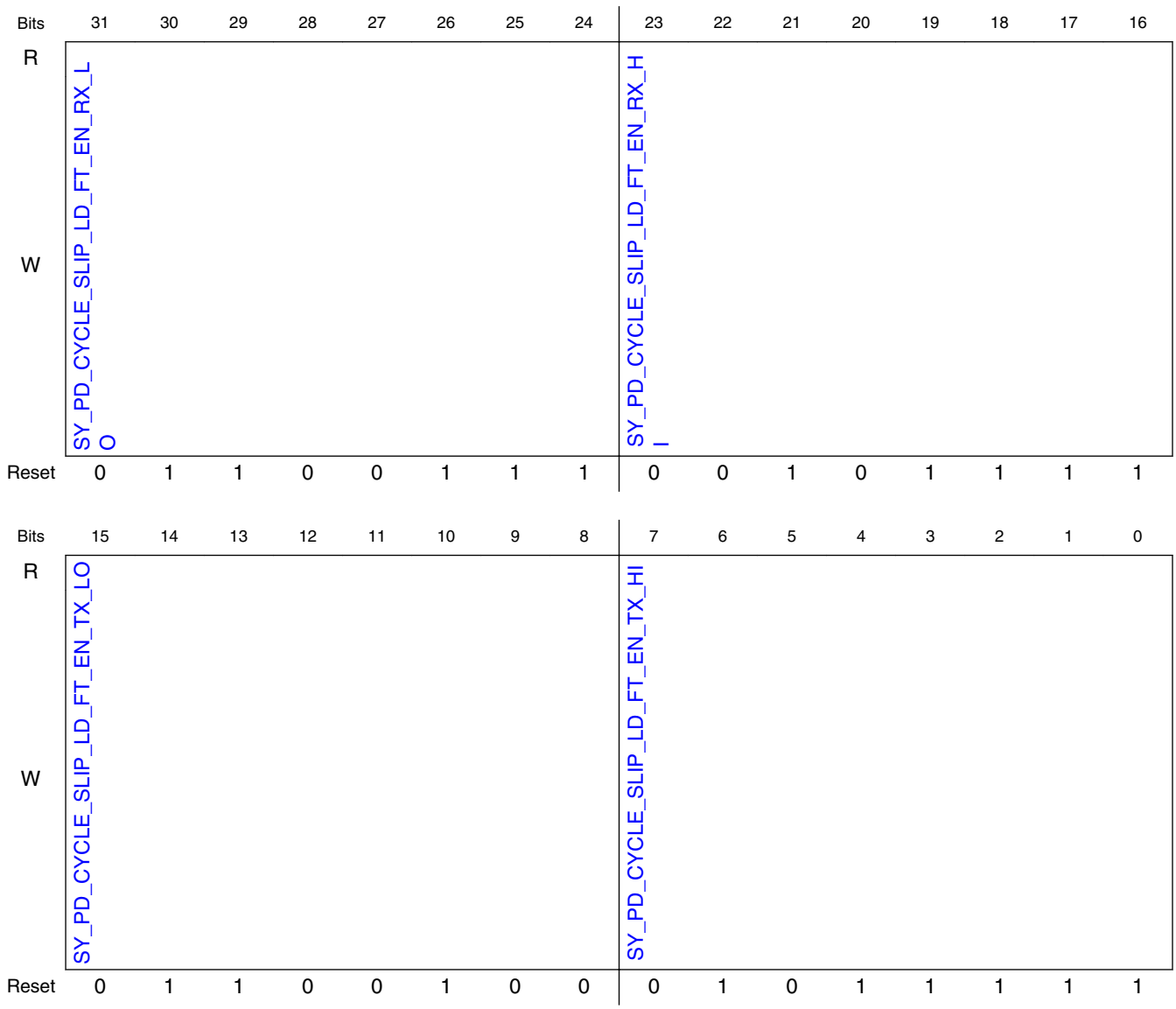
Field	Function
31-24 PLL_LOOP_IS_OPEN_RX_LO	De-assertion time setting for PLL_LOOP_IS_OPEN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_LOOP_IS_OPEN signal or group will transition from HI to LO.
23-16 PLL_LOOP_IS_OPEN_RX_HI	Assertion time setting for PLL_LOOP_IS_OPEN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the PLL_LOOP_IS_OPEN signal or group will transition from LO to HI.
15-8 PLL_LOOP_IS_OPEN_TX_LO	De-assertion time setting for PLL_LOOP_IS_OPEN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_LOOP_IS_OPEN signal or group will transition from HI to LO.
7-0 PLL_LOOP_IS_OPEN_TX_HI	Assertion time setting for PLL_LOOP_IS_OPEN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the PLL_LOOP_IS_OPEN signal or group will transition from LO to HI.

A.2.4.26 TSM_TIMING14 (TIMING14)

A.2.4.26.1 Offset

Register	Offset
TIMING14	68h

A.2.4.26.2 Diagram



A.2.4.26.3 Fields

Field	Function
31-24	De-assertion time setting for SY_PD_CYCLE_SLIP_LD_FT_EN (RX)

Table continues on the next page...

Field	Function
SY_PD_CYCLE_SLIP_LD_FT_EN_RX_LO	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_CYCLE_SLIP_LD_FT_EN signal or group will transition from HI to LO.
23-16 SY_PD_CYCLE_SLIP_LD_FT_EN_RX_HI	Assertion time setting for SY_PD_CYCLE_SLIP_LD_FT_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_CYCLE_SLIP_LD_FT_EN signal or group will transition from LO to HI.
15-8 SY_PD_CYCLE_SLIP_LD_FT_EN_TX_LO	De-assertion time setting for SY_PD_CYCLE_SLIP_LD_FT_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_CYCLE_SLIP_LD_FT_EN signal or group will transition from HI to LO.
7-0 SY_PD_CYCLE_SLIP_LD_FT_EN_TX_HI	Assertion time setting for SY_PD_CYCLE_SLIP_LD_FT_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_CYCLE_SLIP_LD_FT_EN signal or group will transition from LO to HI.

A.2.4.27 TSM_TIMING15 (TIMING15)

A.2.4.27.1 Offset

Register	Offset
TIMING15	6Ch

A.2.4.27.2 Diagram

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

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

A.2.4.27.3 Fields

Field	Function
31-24 SY_VCO_EN_RX_LO	De-assertion time setting for SY_VCO_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_VCO_EN signal or group will transition from HI to LO.
23-16 SY_VCO_EN_RX_HI	Assertion time setting for SY_VCO_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_VCO_EN signal or group will transition from LO to HI.
15-8 SY_VCO_EN_TX_LO	De-assertion time setting for SY_VCO_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_VCO_EN signal or group will transition from HI to LO.
7-0 SY_VCO_EN_TX_HI	Assertion time setting for SY_VCO_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_VCO_EN signal or group will transition from LO to HI.

A.2.4.28 TSM_TIMING16 (TIMING16)

A.2.4.28.1 Offset

Register	Offset
TIMING16	70h

A.2.4.28.2 Diagram

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

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

A.2.4.28.3 Fields

Field	Function
31-24 SY_LO_RX_BUF_EN_RX_LO	De-assertion time setting for SY_LO_RX_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_RX_BUF_EN signal or group will transition from HI to LO.
23-16 SY_LO_RX_BUF_EN_RX_HI	Assertion time setting for SY_LO_RX_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_RX_BUF_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.29 TSM_TIMING17 (TIMING17)

A.2.4.29.1 Offset

Register	Offset
TIMING17	74h

A.2.4.29.2 Diagram

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

A.2.4.29.3 Fields

Field	Function
31-16 —	Reserved
15-8	De-assertion time setting for SY_LO_TX_BUF_EN (TX)

Table continues on the next page...

Field	Function
SY_LO_TX_BUF_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_TX_BUF_EN signal or group will transition from HI to LO.
7-0 SY_LO_TX_BUF_EN_TX_HI	Assertion time setting for SY_LO_TX_BUF_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_TX_BUF_EN signal or group will transition from LO to HI.

A.2.4.30 TSM_TIMING18 (TIMING18)

A.2.4.30.1 Offset

Register	Offset
TIMING18	78h

A.2.4.30.2 Diagram

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

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

A.2.4.30.3 Fields

Field	Function
31-24 SY_DIVN_EN_RX_LO	De-assertion time setting for SY_DIVN_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_DIVN_EN signal or group will transition from HI to LO.
23-16 SY_DIVN_EN_RX_HI	Assertion time setting for SY_DIVN_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_DIVN_EN signal or group will transition from LO to HI.
15-8 SY_DIVN_EN_TX_LO	De-assertion time setting for SY_DIVN_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_DIVN_EN signal or group will transition from HI to LO.

Table continues on the next page...

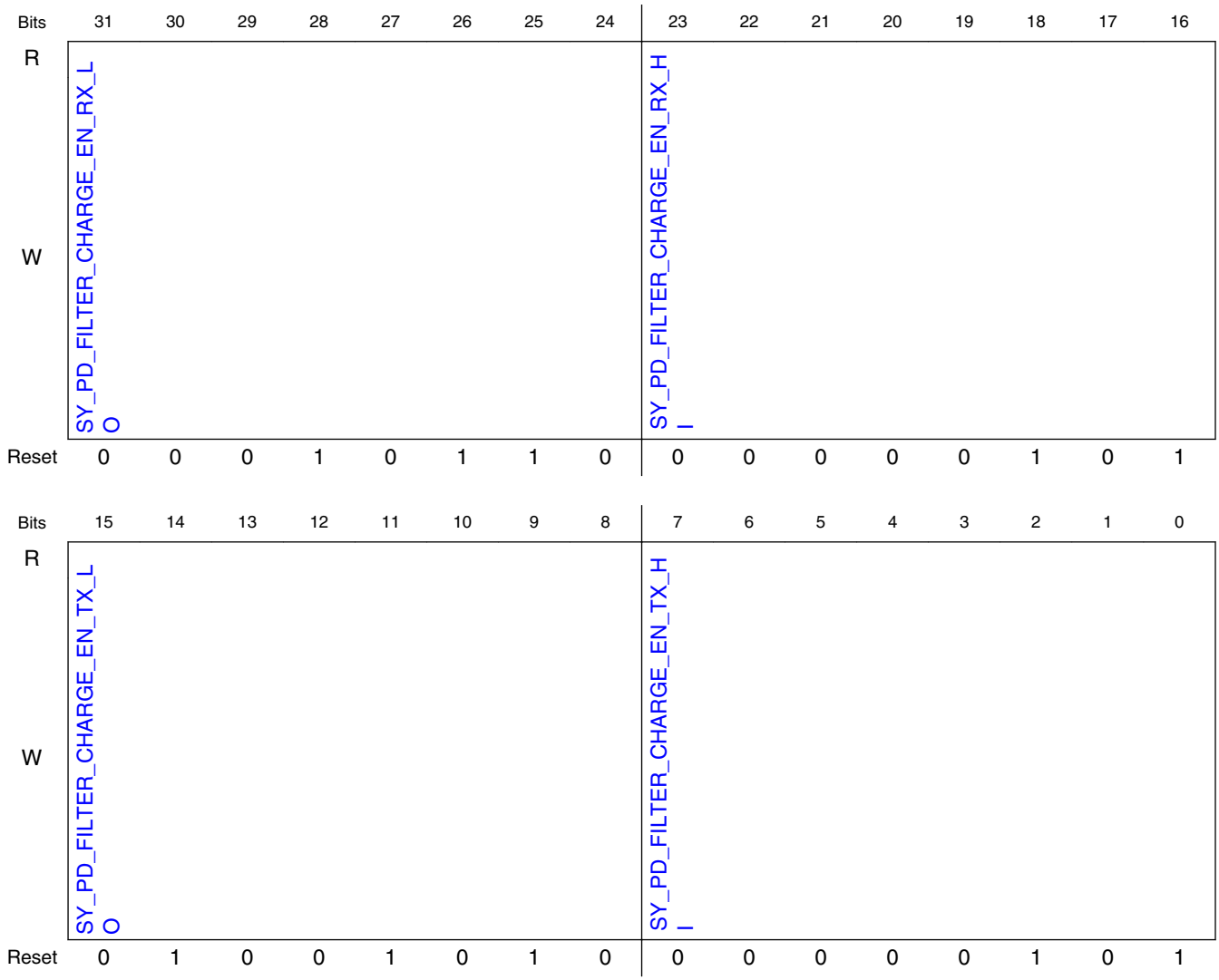
Field	Function
7-0 SY_DIVN_EN_TX_HI	Assertion time setting for SY_DIVN_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_DIVN_EN signal or group will transition from LO to HI.

A.2.4.31 TSM_TIMING19 (TIMING19)

A.2.4.31.1 Offset

Register	Offset
TIMING19	7Ch

A.2.4.31.2 Diagram



A.2.4.31.3 Fields

Field	Function
31-24 SY_PD_FILTER_CHARGE_EN_RX_LO	De-assertion time setting for SY_PD_FILTER_CHARGE_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_FILTER_CHARGE_EN signal or group will transition from HI to LO.
23-16 SY_PD_FILTER_CHARGE_EN_RX_HI	Assertion time setting for SY_PD_FILTER_CHARGE_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_FILTER_CHARGE_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for SY_PD_FILTER_CHARGE_EN (TX)

Table continues on the next page...

Field	Function
SY_PD_FILTER_CHARGE_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_FILTER_CHARGE_EN signal or group will transition from HI to LO.
7-0 SY_PD_FILTER_CHARGE_EN_TX_HI	Assertion time setting for SY_PD_FILTER_CHARGE_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_FILTER_CHARGE_EN signal or group will transition from LO to HI.

A.2.4.32 TSM_TIMING20 (TIMING20)

A.2.4.32.1 Offset

Register	Offset
TIMING20	80h

A.2.4.32.2 Diagram

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

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

A.2.4.32.3 Fields

Field	Function
31-24 SY_PD_EN_RX_LO	De-assertion time setting for SY_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_EN signal or group will transition from HI to LO.
23-16 SY_PD_EN_RX_HI	Assertion time setting for SY_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_PD_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for SY_PD_EN (TX)

Table continues on the next page...

Field	Function
SY_PD_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_EN signal or group will transition from HI to LO.
7-0	Assertion time setting for SY_PD_EN (TX)
SY_PD_EN_TX_HI	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_PD_EN signal or group will transition from LO to HI.

A.2.4.33 TSM_TIMING21 (TIMING21)

A.2.4.33.1 Offset

Register	Offset
TIMING21	84h

A.2.4.33.2 Diagram

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

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

A.2.4.33.3 Fields

Field	Function
31-24	De-assertion time setting for SY_LO_DIVN_EN (RX)
SY_LO_DIVN_EN_RX_LO	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_DIVN_EN signal or group will transition from HI to LO.
23-16	Assertion time setting for SY_LO_DIVN_EN (RX)
SY_LO_DIVN_EN_RX_HI	This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_DIVN_EN signal or group will transition from LO to HI.
15-8	De-assertion time setting for SY_LO_DIVN_EN (TX)
SY_LO_DIVN_EN_TX_LO	This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_DIVN_EN signal or group will transition from HI to LO.

Table continues on the next page...

Field	Function
7-0 SY_LO_DIVN_EN_TX_HI	Assertion time setting for SY_LO_DIVN_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_DIVN_EN signal or group will transition from LO to HI.

A.2.4.34 TSM_TIMING22 (TIMING22)

A.2.4.34.1 Offset

Register	Offset
TIMING22	88h

A.2.4.34.2 Diagram

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

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

A.2.4.34.3 Fields

Field	Function
31-24 SY_LO_RX_EN_RX_LO	De-assertion time setting for SY_LO_RX_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_RX_EN signal or group will transition from HI to LO.
23-16 SY_LO_RX_EN_RX_HI	Assertion time setting for SY_LO_RX_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_LO_RX_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.35 TSM_TIMING23 (TIMING23)

A.2.4.35.1 Offset

Register	Offset
TIMING23	8Ch

A.2.4.35.2 Diagram

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

A.2.4.35.3 Fields

Field	Function
31-16 —	Reserved
15-8 SY_LO_TX_EN_TX_LO	De-assertion time setting for SY_LO_TX_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_TX_EN signal or group will transition from HI to LO.
7-0 SY_LO_TX_EN_TX_HI	Assertion time setting for SY_LO_TX_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_LO_TX_EN signal or group will transition from LO to HI.

A.2.4.36 TSM_TIMING24 (TIMING24)

A.2.4.36.1 Offset

Register	Offset
TIMING24	90h

A.2.4.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SY_DIVN_CAL_EN_RX_LO								SY_DIVN_CAL_EN_RX_HI							
Reset	0	0	0	1	0	1	1	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	SY_DIVN_CAL_EN_TX_LO								SY_DIVN_CAL_EN_TX_HI							
Reset	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

A.2.4.36.3 Fields

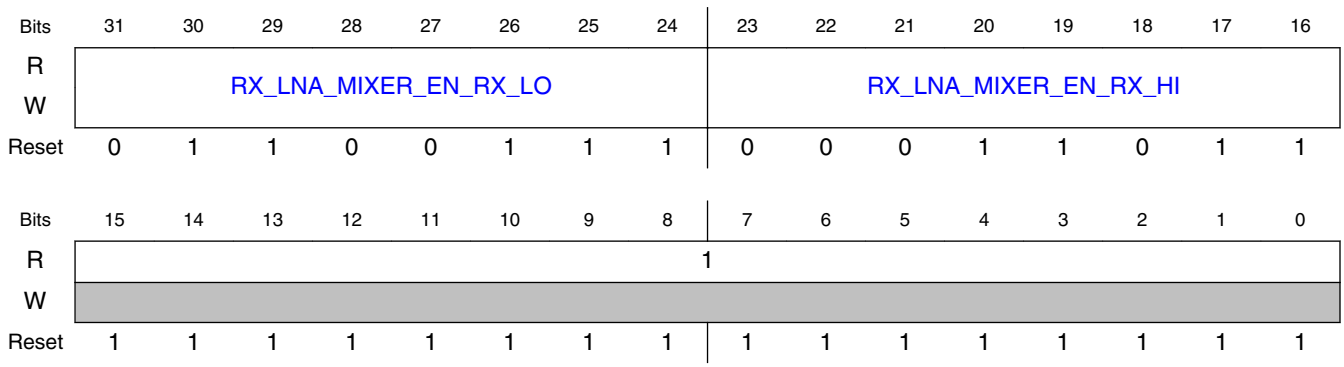
Field	Function
31-24 SY_DIVN_CAL_EN_RX_LO	De-assertion time setting for SY_DIVN_CAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_DIVN_CAL_EN signal or group will transition from HI to LO.
23-16 SY_DIVN_CAL_EN_RX_HI	Assertion time setting for SY_DIVN_CAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the SY_DIVN_CAL_EN signal or group will transition from LO to HI.
15-8 SY_DIVN_CAL_EN_TX_LO	De-assertion time setting for SY_DIVN_CAL_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_DIVN_CAL_EN signal or group will transition from HI to LO.
7-0 SY_DIVN_CAL_EN_TX_HI	Assertion time setting for SY_DIVN_CAL_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the SY_DIVN_CAL_EN signal or group will transition from LO to HI.

A.2.4.37 TSM_TIMING25 (TIMING25)

A.2.4.37.1 Offset

Register	Offset
TIMING25	94h

A.2.4.37.2 Diagram



A.2.4.37.3 Fields

Field	Function
31-24 RX_LNA_MIXER_EN_RX_LO	De-assertion time setting for RX_LNA_MIXER_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_LNA_MIXER_EN signal or group will transition from HI to LO.
23-16 RX_LNA_MIXER_EN_RX_HI	Assertion time setting for RX_LNA_MIXER_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_LNA_MIXER_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.38 TSM_TIMING26 (TIMING26)

A.2.4.38.1 Offset

Register	Offset
TIMING26	98h

A.2.4.38.2 Diagram

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

A.2.4.38.3 Fields

Field	Function
31-16 —	Reserved
15-8 TX_PA_EN_TX_LO	De-assertion time setting for TX_PA_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_PA_EN signal or group will transition from HI to LO.
7-0 TX_PA_EN_TX_HI	Assertion time setting for TX_PA_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_PA_EN signal or group will transition from LO to HI.

A.2.4.39 TSM_TIMING27 (TIMING27)

A.2.4.39.1 Offset

Register	Offset
TIMING27	9Ch

A.2.4.39.2 Diagram

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

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

A.2.4.39.3 Fields

Field	Function
31-24 RX_ADC_I_Q_EN_RX_LO	De-assertion time setting for RX_ADC_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_ADC_I_Q_EN signal or group will transition from HI to LO.
23-16 RX_ADC_I_Q_EN_RX_HI	Assertion time setting for RX_ADC_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_ADC_I_Q_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.40 TSM_TIMING28 (TIMING28)

A.2.4.40.1 Offset

Register	Offset
TIMING28	A0h

A.2.4.40.2 Diagram

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

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

A.2.4.40.3 Fields

Field	Function
31-24 RX_ADC_RESET_EN_RX_LO	De-assertion time setting for RX_ADC_RESET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_ADC_RESET_EN signal or group will transition from HI to LO.
23-16 RX_ADC_RESET_EN_RX_HI	Assertion time setting for RX_ADC_RESET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_ADC_RESET_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.41 TSM_TIMING29 (TIMING29)

A.2.4.41.1 Offset

Register	Offset
TIMING29	A4h

A.2.4.41.2 Diagram

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

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

A.2.4.41.3 Fields

Field	Function
31-24 RX_BBA_I_Q_EN_RX_LO	De-assertion time setting for RX_BBA_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_I_Q_EN signal or group will transition from HI to LO.
23-16 RX_BBA_I_Q_EN_RX_HI	Assertion time setting for RX_BBA_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_I_Q_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.42 TSM_TIMING30 (TIMING30)

A.2.4.42.1 Offset

Register	Offset
TIMING30	A8h

A.2.4.42.2 Diagram

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

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

A.2.4.42.3 Fields

Field	Function
31-24 RX_BBA_PDET_EN_RX_LO	De-assertion time setting for RX_BBA_PDET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_PDET_EN signal or group will transition from HI to LO.
23-16 RX_BBA_PDET_EN_RX_HI	Assertion time setting for RX_BBA_PDET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_PDET_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.43 TSM_TIMING31 (TIMING31)

A.2.4.43.1 Offset

Register	Offset
TIMING31	ACh

A.2.4.43.2 Diagram

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

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

A.2.4.43.3 Fields

Field	Function
31-24 RX_BBA_TZA_DCOC_EN_RX_LO	De-assertion time setting for RX_BBA_TZA_DCOC_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_TZA_DCOC_EN signal or group will transition from HI to LO.
23-16 RX_BBA_TZA_DCOC_EN_RX_HI	Assertion time setting for RX_BBA_TZA_DCOC_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_BBA_TZA_DCOC_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.44 TSM_TIMING32 (TIMING32)

A.2.4.44.1 Offset

Register	Offset
TIMING32	B0h

A.2.4.44.2 Diagram

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

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

A.2.4.44.3 Fields

Field	Function
31-24 RX_TZA_I_Q_EN_RX_LO	De-assertion time setting for RX_TZA_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_TZA_I_Q_EN signal or group will transition from HI to LO.
23-16 RX_TZA_I_Q_EN_RX_HI	Assertion time setting for RX_TZA_I_Q_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_TZA_I_Q_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.45 TSM_TIMING33 (TIMING33)

A.2.4.45.1 Offset

Register	Offset
TIMING33	B4h

A.2.4.45.2 Diagram

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

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

A.2.4.45.3 Fields

Field	Function
31-24 RX_TZA_PDET_EN_RX_LO	De-assertion time setting for RX_TZA_PDET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_TZA_PDET_EN signal or group will transition from HI to LO.
23-16 RX_TZA_PDET_EN_RX_HI	Assertion time setting for RX_TZA_PDET_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_TZA_PDET_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.46 TSM_TIMING34 (TIMING34)

A.2.4.46.1 Offset

Register	Offset
TIMING34	B8h

A.2.4.46.2 Diagram

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

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

A.2.4.46.3 Fields

Field	Function
31-24 PLL_DIG_EN_RX_LO	De-assertion time setting for PLL_DIG_EN (RX) 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 (RX) 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	De-assertion time setting for PLL_DIG_EN (TX) 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.
7-0 PLL_DIG_EN_TX_HI	Assertion time setting for PLL_DIG_EN (TX) 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.

A.2.4.47 TSM_TIMING35 (TIMING35)

A.2.4.47.1 Offset

Register	Offset
TIMING35	BCh

A.2.4.47.2 Diagram

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

A.2.4.47.3 Fields

Field	Function
31-16 —	Reserved
15-8 TX_DIG_EN_TX_LO	De-assertion time setting for TX_DIG_EN (TX) 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.
7-0 TX_DIG_EN_TX_HI	Assertion time setting for TX_DIG_EN (TX) 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.

A.2.4.48 TSM_TIMING36 (TIMING36)

A.2.4.48.1 Offset

Register	Offset
TIMING36	C0h

A.2.4.48.2 Diagram

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

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

A.2.4.48.3 Fields

Field	Function
31-24 RX_DIG_EN_RX_LO	De-assertion time setting for RX_DIG_EN (RX) 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 (RX) 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.
15-0 —	Reserved

A.2.4.49 TSM_TIMING37 (TIMING37)

A.2.4.49.1 Offset

Register	Offset
TIMING37	C4h

A.2.4.49.2 Diagram

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

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

A.2.4.49.3 Fields

Field	Function
31-24 RX_INIT_RX_LO	De-assertion time setting for RX_INIT (RX) 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 (RX) 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.
15-0 —	Reserved

A.2.4.50 TSM_TIMING38 (TIMING38)

A.2.4.50.1 Offset

Register	Offset
TIMING38	C8h

A.2.4.50.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SIGMA_DELTA_EN_RX_LO								SIGMA_DELTA_EN_RX_HI							
Reset	0	1	1	0	0	1	1	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	SIGMA_DELTA_EN_TX_LO								SIGMA_DELTA_EN_TX_HI							
Reset	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0

A.2.4.50.3 Fields

Field	Function
31-24 SIGMA_DELTA_EN_RX_LO	De-assertion time setting for SIGMA_DELTA_EN (RX) 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 (RX) 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	De-assertion time setting for SIGMA_DELTA_EN (TX) 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.
7-0 SIGMA_DELTA_EN_TX_HI	Assertion time setting for SIGMA_DELTA_EN (TX) 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.

A.2.4.51 TSM_TIMING39 (TIMING39)

A.2.4.51.1 Offset

Register	Offset
TIMING39	CCh

A.2.4.51.2 Diagram

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

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

A.2.4.51.3 Fields

Field	Function
31-24 RX_PHY_EN_RX_LO	De-assertion time setting for RX_PHY_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_PHY_EN signal or group will transition from HI to LO.
23-16 RX_PHY_EN_RX_HI	Assertion time setting for RX_PHY_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RX_PHY_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.52 TSM_TIMING40 (TIMING40)

A.2.4.52.1 Offset

Register	Offset
TIMING40	D0h

A.2.4.52.2 Diagram

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

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

A.2.4.52.3 Fields

Field	Function
31-24 DCOC_EN_RX_LO	De-assertion time setting for DCOC_EN (RX) 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 (RX) 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.
15-0 —	Reserved

A.2.4.53 TSM_TIMING41 (TIMING41)

A.2.4.53.1 Offset

Register	Offset
TIMING41	D4h

A.2.4.53.2 Diagram

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

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

A.2.4.53.3 Fields

Field	Function
31-24 DCOC_INIT_RX_LO	De-assertion time setting for DCOC_INIT (RX) 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 (RX) 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.
15-0 —	Reserved

A.2.4.54 TSM_TIMING42 (TIMING42)

A.2.4.54.1 Offset

Register	Offset
TIMING42	D8h

A.2.4.54.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SAR_ADC_TRIG_EN_RX_LO								SAR_ADC_TRIG_EN_RX_HI							
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	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

A.2.4.54.3 Fields

Field	Function
31-24 SAR_ADC_TRIG_EN_RX_LO	De-assertion time setting for SAR_ADC_TRIG_EN (RX) 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 (RX) 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	De-assertion time setting for SAR_ADC_TRIG_EN (TX) 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.
7-0 SAR_ADC_TRIG_EN_TX_HI	Assertion time setting for SAR_ADC_TRIG_EN (TX) 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.

A.2.4.55 TSM_TIMING43 (TIMING43)

A.2.4.55.1 Offset

Register	Offset
TIMING43	DCh

A.2.4.55.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSM_SPARE0_EN_RX_LO								TSM_SPARE0_EN_RX_HI							
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	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

A.2.4.55.3 Fields

Field	Function
31-24 TSM_SPARE0_EN_RX_LO	De-assertion time setting for TSM_SPARE0_EN (RX) 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 (RX) 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	De-assertion time setting for TSM_SPARE0_EN (TX) 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.
7-0 TSM_SPARE0_EN_TX_HI	Assertion time setting for TSM_SPARE0_EN (TX) 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.

A.2.4.56 TSM_TIMING44 (TIMING44)

A.2.4.56.1 Offset

Register	Offset
TIMING44	E0h

A.2.4.56.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSM_SPARE1_EN_RX_LO								TSM_SPARE1_EN_RX_HI							
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	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

A.2.4.56.3 Fields

Field	Function
31-24 TSM_SPARE1_EN_RX_LO	De-assertion time setting for TSM_SPARE1_EN (RX) 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 (RX) 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	De-assertion time setting for TSM_SPARE1_EN (TX) 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.
7-0 TSM_SPARE1_EN_TX_HI	Assertion time setting for TSM_SPARE1_EN (TX) 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.

A.2.4.57 TSM_TIMING45 (TIMING45)

A.2.4.57.1 Offset

Register	Offset
TIMING45	E4h

A.2.4.57.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSM_SPARE2_EN_RX_LO								TSM_SPARE2_EN_RX_HI							
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	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

A.2.4.57.3 Fields

Field	Function
31-24 TSM_SPARE2_EN_RX_LO	De-assertion time setting for TSM_SPARE2_EN (RX) 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 (RX) 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	De-assertion time setting for TSM_SPARE2_EN (TX) 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.
7-0 TSM_SPARE2_EN_TX_HI	Assertion time setting for TSM_SPARE2_EN (TX) 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.

A.2.4.58 TSM_TIMING46 (TIMING46)

A.2.4.58.1 Offset

Register	Offset
TIMING46	E8h

A.2.4.58.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSM_SPARE3_EN_RX_LO								TSM_SPARE3_EN_RX_HI							
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	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

A.2.4.58.3 Fields

Field	Function
31-24 TSM_SPARE3_EN_RX_LO	De-assertion time setting for TSM_SPARE3_EN (RX) 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 (RX) 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	De-assertion time setting for TSM_SPARE3_EN (TX) 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.
7-0 TSM_SPARE3_EN_TX_HI	Assertion time setting for TSM_SPARE3_EN (TX) 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.

A.2.4.59 TSM_TIMING47 (TIMING47)

A.2.4.59.1 Offset

Register	Offset
TIMING47	ECh

A.2.4.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO0_TRIG_EN_RX_LO								GPIO0_TRIG_EN_RX_HI							
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	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

A.2.4.59.3 Fields

Field	Function
31-24 GPIO0_TRIG_EN_RX_LO	De-assertion time setting for GPIO0_TRIG_EN (RX) 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 (RX) 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	De-assertion time setting for GPIO0_TRIG_EN (TX) 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.
7-0 GPIO0_TRIG_EN_TX_HI	Assertion time setting for GPIO0_TRIG_EN (TX) 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.

A.2.4.60 TSM_TIMING48 (TIMING48)

A.2.4.60.1 Offset

Register	Offset
TIMING48	F0h

A.2.4.60.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO1_TRIG_EN_RX_LO								GPIO1_TRIG_EN_RX_HI							
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	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

A.2.4.60.3 Fields

Field	Function
31-24 GPIO1_TRIG_EN_RX_LO	De-assertion time setting for GPIO1_TRIG_EN (RX) 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 (RX) 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	De-assertion time setting for GPIO1_TRIG_EN (TX) 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.
7-0 GPIO1_TRIG_EN_TX_HI	Assertion time setting for GPIO1_TRIG_EN (TX) 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.

A.2.4.61 TSM_TIMING49 (TIMING49)

A.2.4.61.1 Offset

Register	Offset
TIMING49	F4h

A.2.4.61.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO2_TRIG_EN_RX_LO								GPIO2_TRIG_EN_RX_HI							
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	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

A.2.4.61.3 Fields

Field	Function
31-24 GPIO2_TRIG_EN_RX_LO	De-assertion time setting for GPIO2_TRIG_EN (RX) 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 (RX) 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	De-assertion time setting for GPIO2_TRIG_EN (TX) 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.
7-0 GPIO2_TRIG_EN_TX_HI	Assertion time setting for GPIO2_TRIG_EN (TX) 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.

A.2.4.62 TSM_TIMING50 (TIMING50)

A.2.4.62.1 Offset

Register	Offset
TIMING50	F8h

A.2.4.62.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO3_TRIG_EN_RX_LO								GPIO3_TRIG_EN_RX_HI							
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	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

A.2.4.62.3 Fields

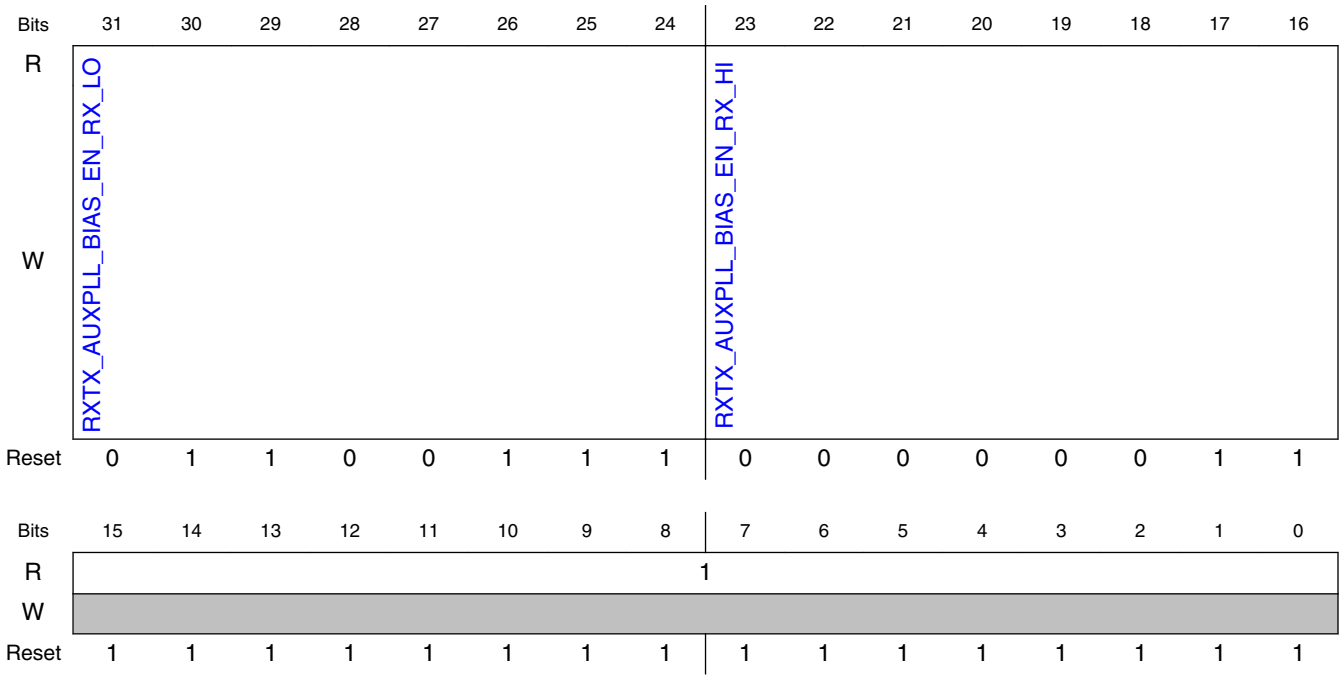
Field	Function
31-24 GPIO3_TRIG_EN_RX_LO	De-assertion time setting for GPIO3_TRIG_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the GPIO3_TRIG_EN signal or group will transition from HI to LO.
23-16 GPIO3_TRIG_EN_RX_HI	Assertion time setting for GPIO3_TRIG_EN (RX) 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	De-assertion time setting for GPIO3_TRIG_EN (TX) 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.
7-0 GPIO3_TRIG_EN_TX_HI	Assertion time setting for GPIO3_TRIG_EN (TX) 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.

A.2.4.63 TSM_TIMING51 (TIMING51)

A.2.4.63.1 Offset

Register	Offset
TIMING51	FCh

A.2.4.63.2 Diagram



A.2.4.63.3 Fields

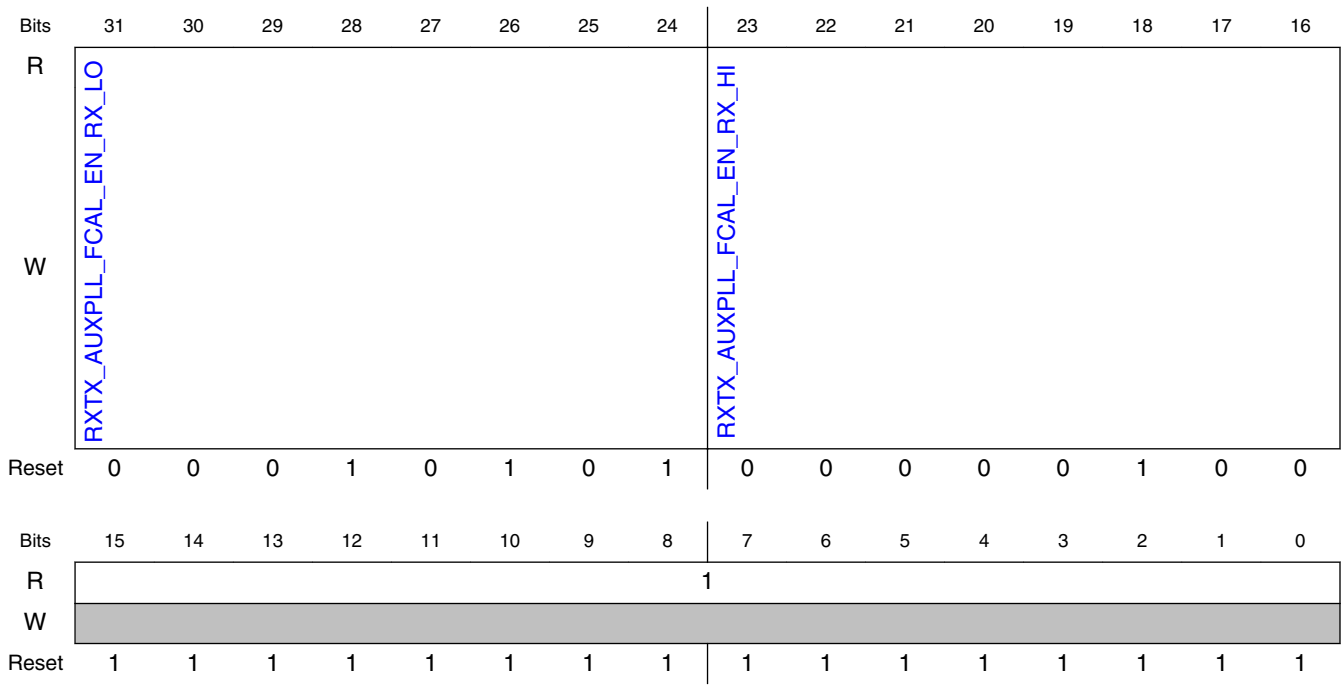
Field	Function
31-24 RXTX_AUXPLL_BIAS_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_BIAS_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_BIAS_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_BIAS_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_BIAS_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_BIAS_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.64 TSM_TIMING52 (TIMING52)

A.2.4.64.1 Offset

Register	Offset
TIMING52	100h

A.2.4.64.2 Diagram



A.2.4.64.3 Fields

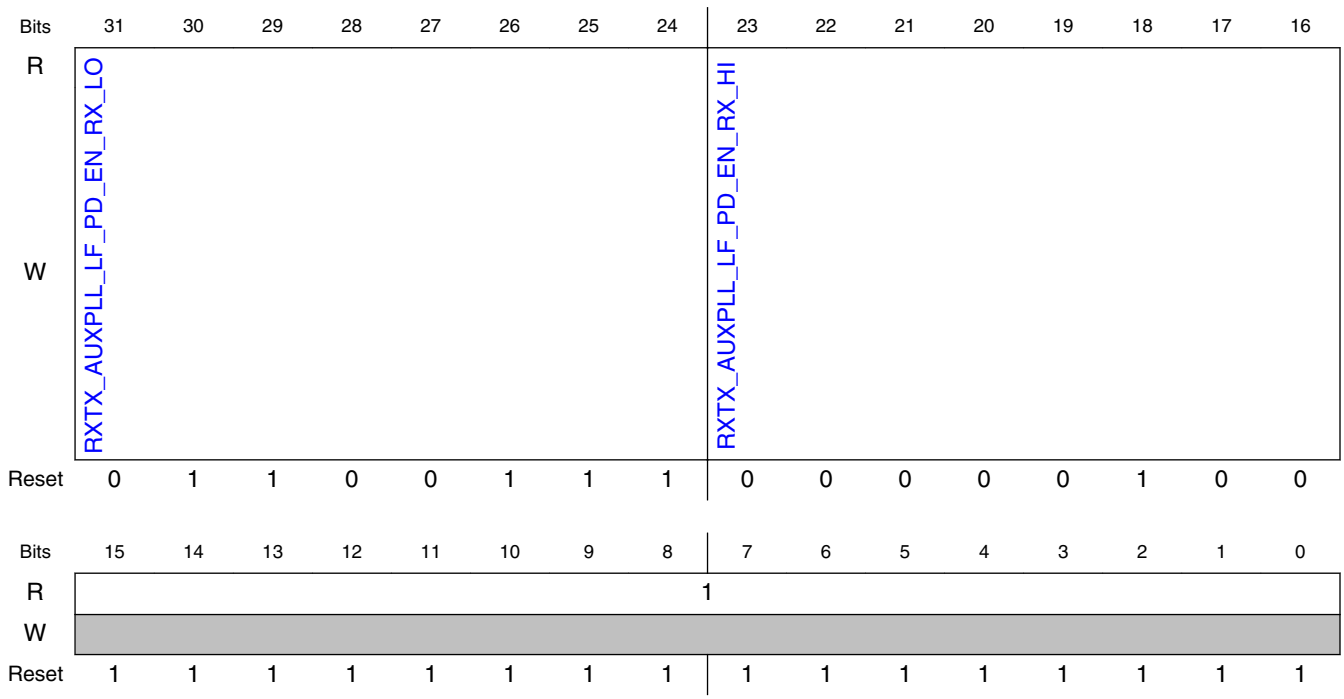
Field	Function
31-24 RXTX_AUXPLL_FCAL_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_FCAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_FCAL_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_FCAL_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_FCAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_FCAL_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.65 TSM_TIMING53 (TIMING53)

A.2.4.65.1 Offset

Register	Offset
TIMING53	104h

A.2.4.65.2 Diagram



A.2.4.65.3 Fields

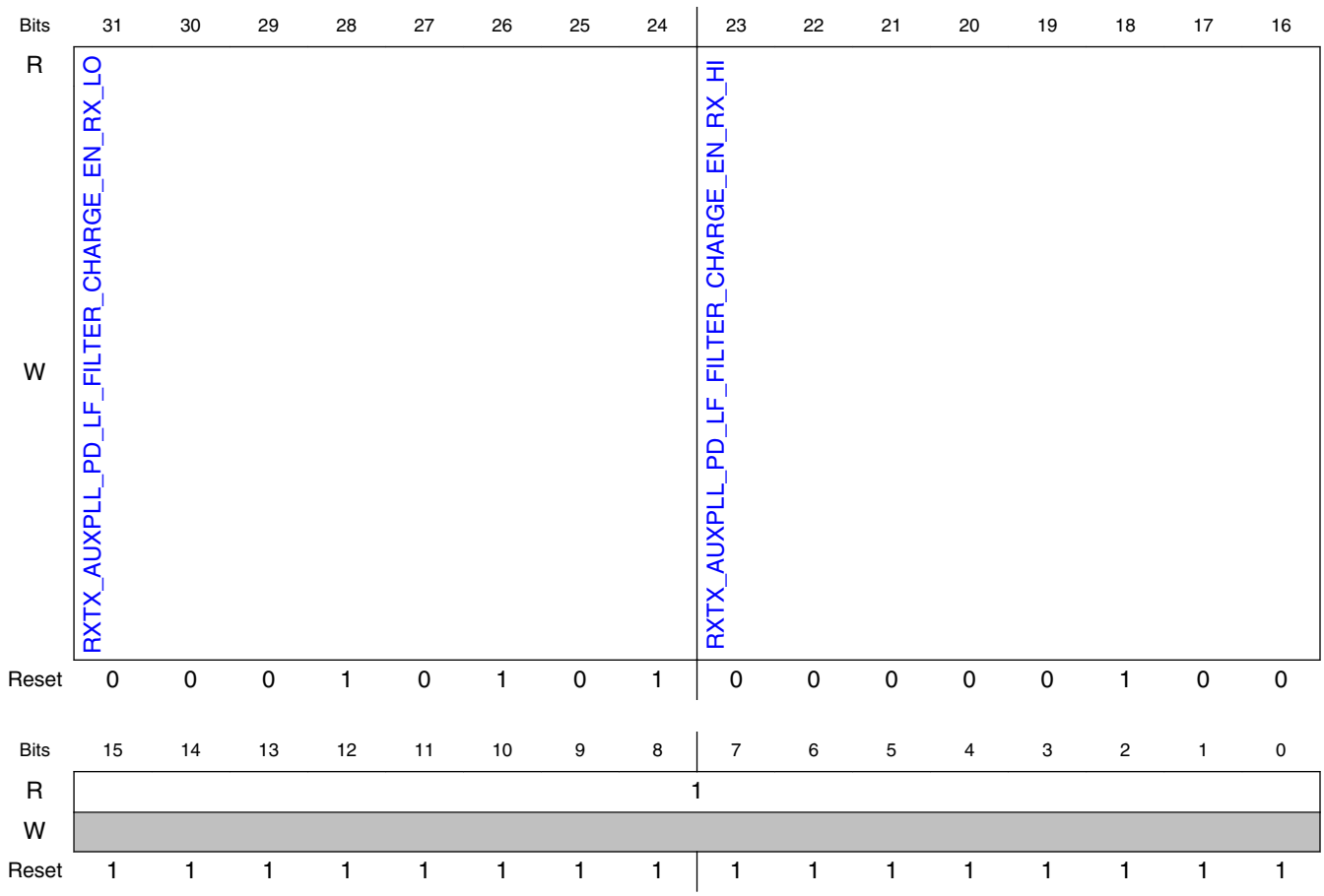
Field	Function
31-24 RXTX_AUXPLL_LF_PD_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_LF_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_LF_PD_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_LF_PD_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_LF_PD_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_LF_PD_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.66 TSM_TIMING54 (TIMING54)

A.2.4.66.1 Offset

Register	Offset
TIMING54	108h

A.2.4.66.2 Diagram



A.2.4.66.3 Fields

Field	Function
31-24 RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN signal or group will transition from HI to LO.

Table continues on the next page...

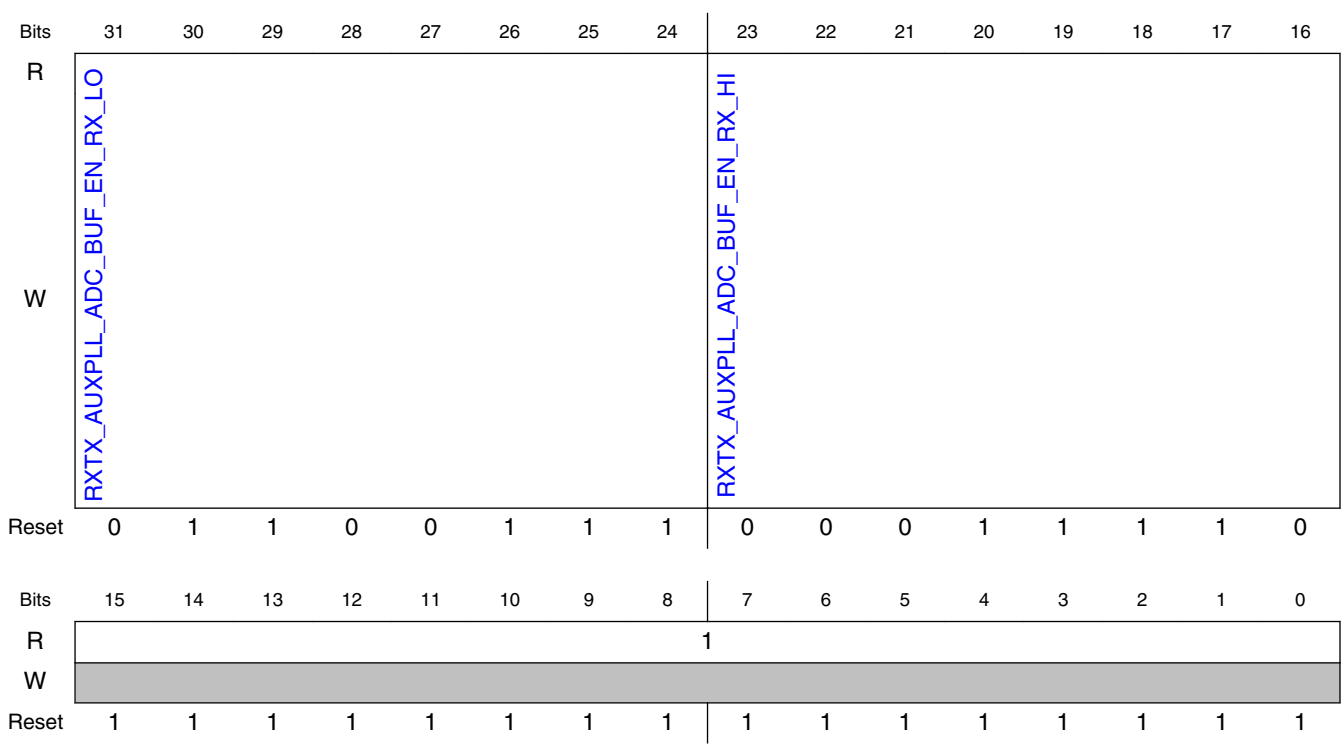
Field	Function
23-16 RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.67 TSM_TIMING55 (TIMING55)

A.2.4.67.1 Offset

Register	Offset
TIMING55	10Ch

A.2.4.67.2 Diagram



A.2.4.67.3 Fields

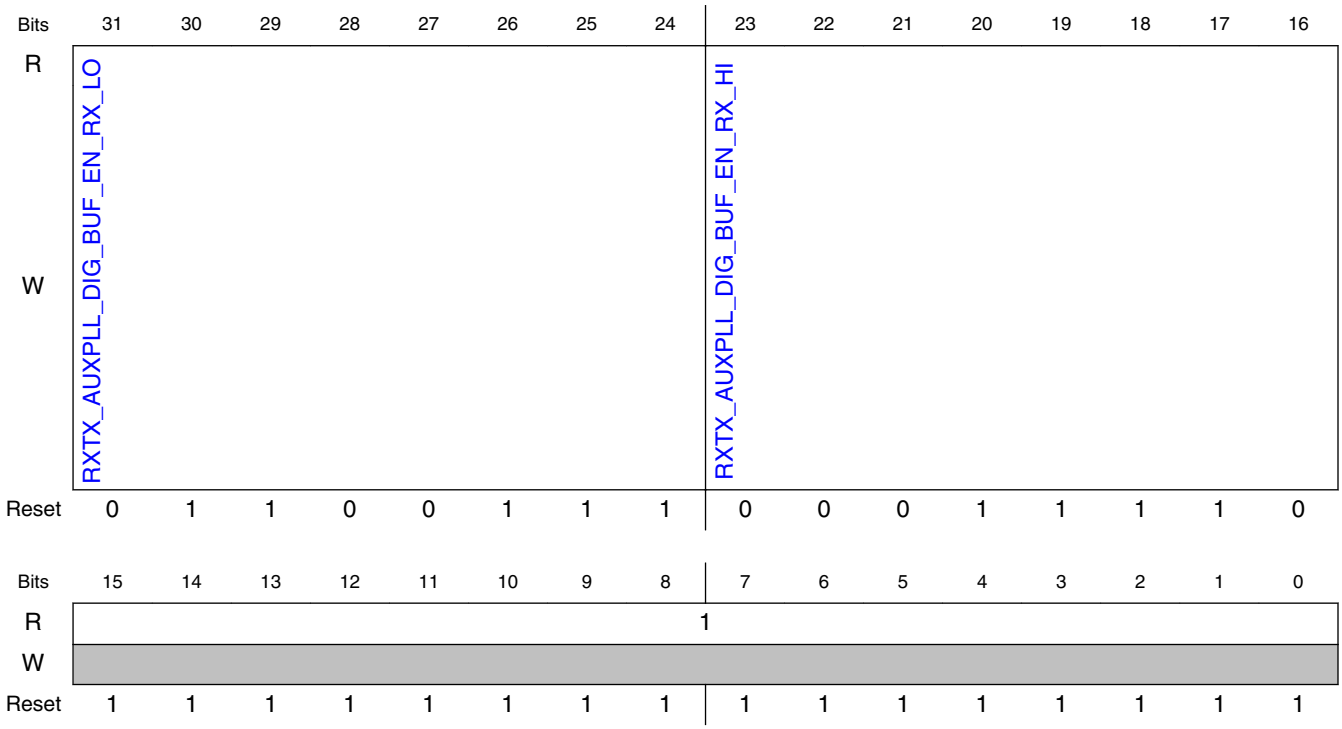
Field	Function
31-24 RXTX_AUXPLL_ADC_BUF_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_ADC_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_ADC_BUF_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_ADC_BUF_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_ADC_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_ADC_BUF_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.68 TSM_TIMING56 (TIMING56)

A.2.4.68.1 Offset

Register	Offset
TIMING56	110h

A.2.4.68.2 Diagram



A.2.4.68.3 Fields

Field	Function
31-24 RXTX_AUXPLL_DIG_BUF_EN_RX_LO	De-assertion time setting for RXTX_AUXPLL_DIG_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_DIG_BUF_EN signal or group will transition from HI to LO.
23-16 RXTX_AUXPLL_DIG_BUF_EN_RX_HI	Assertion time setting for RXTX_AUXPLL_DIG_BUF_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_AUXPLL_DIG_BUF_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.69 TSM_TIMING57 (TIMING57)

A.2.4.69.1 Offset

Register	Offset
TIMING57	114h

A.2.4.69.2 Diagram

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

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

A.2.4.69.3 Fields

Field	Function
31-24 RXTX_RCCAL_EN_RX_LO	De-assertion time setting for RXTX_RCCAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_RCCAL_EN signal or group will transition from HI to LO.
23-16 RXTX_RCCAL_EN_RX_HI	Assertion time setting for RXTX_RCCAL_EN (RX) This field sets the point during a TSM RX sequence (the tsm_count[7:0] value) at which the RXTX_RCCAL_EN signal or group will transition from LO to HI.
15-0 —	Reserved

A.2.4.70 TSM_TIMING58 (TIMING58)

A.2.4.70.1 Offset

Register	Offset
TIMING58	118h

A.2.4.70.2 Diagram

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

A.2.4.70.3 Fields

Field	Function
31-16 —	Reserved
15-8 TX_HPM_DAC_EN_TX_LO	De-assertion time setting for TX_HPM_DAC_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_HPM_DAC_EN signal or group will transition from HI to LO.
7-0 TX_HPM_DAC_EN_TX_HI	Assertion time setting for TX_HPM_DAC_EN (TX) This field sets the point during a TSM TX sequence (the tsm_count[7:0] value) at which the TX_HPM_DAC_EN signal or group will transition from LO to HI.

A.2.4.71 TSM OVERRIDE REGISTER 0 (OVRD0)

A.2.4.71.1 Offset

Register	Offset
OVRD0	11Ch

A.2.4.71.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	SY_VCO_EN_OVRD	SY_VCO_EN_OVRD_EN	SY_PD_CYCLE_SLIP_LD_EN_OVRD	SY_PD_CYCLE_SLIP_LD_EN_OVRD_EN	PLL_LOOP_IS_OPEN_OVRD	PLL_LOOP_IS_OPEN_OVRD_EN	BB_XTAL_AUXPLL_REF_CLK_EN_OVRD	BB_XTAL_AUXPLL_REF_CLK_EN_OVRD_EN	BB_XTAL_DAC_REF_CLK_EN_OVRD	BB_XTAL_DAC_REF_CLK_EN_OVRD_EN	BB_XTAL_PLL_REF_CLK_EN_OVRD	BB_XTAL_PLL_REF_CLK_EN_OVRD_EN	BB_LDO_VCOLO_FASTCHARGE_EN_OVRD	BB_LDO_VCOLO_FASTCHARGE_EN_OVRD_EN	BB_LDO_VCOLO_BLEED_EN_OVRD	BB_LDO_VCOLO_BLEED_EN_OVRD_EN
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	BB_LDO_FDBK_BLEED_EN_OVRD	BB_LDO_FDBK_BLEED_EN_OVRD_EN	BB_LDO_VTREF_EN_OVRD	BB_LDO_VTREF_EN_OVRD_EN	BB_LDO_VCOLO_EN_OVRD	BB_LDO_VCOLO_EN_OVRD_EN	BB_LDO_FDBK_EN_OVRD	BB_LDO_FDBK_EN_OVRD_EN	BB_LDO_PD_EN_OVRD	BB_LDO_PD_EN_OVRD_EN	BB_LDO_BBA_EN_OVRD	BB_LDO_BBA_EN_OVRD_EN	BB_LDO_ADCDAC_EN_OVRD	BB_LDO_ADCDAC_EN_OVRD_EN	BB_LDO_HF_EN_OVRD	BB_LDO_HF_EN_OVRD_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.4.71.3 Fields

Field	Function
31 SY_VCO_EN_OVRD	Override value for SY_VCO_EN When SY_VCO_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_vco_en". This bit is ignored when SY_VCO_EN_OVRD_EN==0.
30	Override control for SY_VCO_EN 0b - Normal operation.

Table continues on the next page...

Field	Function
SY_VCO_EN_OVRD_EN	1b - Use the state of SY_VCO_EN_OVRD to override the signal "sy_vco_en".
29 SY_PD_CYCLE_SLIP_LD_EN_OVRD	Override value for SY_PD_CYCLE_SLIP_LD_EN When SY_PD_CYCLE_SLIP_LD_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_pd_cycle_slip_ld_en". This bit is ignored when SY_PD_CYCLE_SLIP_LD_EN_OVRD_EN==0.
28 SY_PD_CYCLE_SLIP_LD_EN_OVRD_EN	Override control for SY_PD_CYCLE_SLIP_LD_EN 0b - Normal operation. 1b - Use the state of SY_PD_CYCLE_SLIP_LD_EN_OVRD to override the signal "sy_pd_cycle_slip_ld_en".
27 PLL_LOOP_IS_OPEN_OVRD	Override value for PLL_LOOP_IS_OPEN When PLL_LOOP_IS_OPEN_OVRD_EN=1, this value overrides the mission mode state of the signal "pll_loop_is_open". This bit is ignored when PLL_LOOP_IS_OPEN_OVRD_EN==0.
26 PLL_LOOP_IS_OPEN_OVRD_EN	Override control for PLL_LOOP_IS_OPEN 0b - Normal operation. 1b - Use the state of PLL_LOOP_IS_OPEN_OVRD to override the signal "pll_loop_is_open".
25 BB_XTAL_AUXPLL_REF_CLK_EN_OVRD	Override value for BB_XTAL_AUXPLL_REF_CLK_EN When BB_XTAL_AUXPLL_REF_CLK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_xtal_auxpll_ref_clk_en". This bit is ignored when BB_XTAL_AUXPLL_REF_CLK_EN_OVRD_EN==0.
24 BB_XTAL_AUXPLL_REF_CLK_EN_OVRD_EN	Override control for BB_XTAL_AUXPLL_REF_CLK_EN 0b - Normal operation. 1b - Use the state of BB_XTAL_AUXPLL_REF_CLK_EN_OVRD to override the signal "bb_xtal_auxpll_ref_clk_en".
23 BB_XTAL_DAC_REF_CLK_EN_OVRD	Override value for BB_XTAL_DAC_REF_CLK_EN When BB_XTAL_DAC_REF_CLK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_xtal_dac_ref_clk_en". This bit is ignored when BB_XTAL_DAC_REF_CLK_EN_OVRD_EN==0.
22 BB_XTAL_DAC_REF_CLK_EN_OVRD_EN	Override control for BB_XTAL_DAC_REF_CLK_EN 0b - Normal operation. 1b - Use the state of BB_XTAL_DAC_REF_CLK_EN_OVRD to override the signal "bb_xtal_dac_ref_clk_en".
21 BB_XTAL_PLL_REF_CLK_EN_OVRD	Override value for BB_XTAL_PLL_REF_CLK_EN When BB_XTAL_PLL_REF_CLK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_xtal_pll_ref_clk_en". This bit is ignored when BB_XTAL_PLL_REF_CLK_EN_OVRD_EN==0.
20 BB_XTAL_PLL_REF_CLK_EN_OVRD_EN	Override control for BB_XTAL_PLL_REF_CLK_EN 0b - Normal operation. 1b - Use the state of BB_XTAL_PLL_REF_CLK_EN_OVRD to override the signal "bb_xtal_pll_ref_clk_en".
19 BB_LDO_VCOLO_FASTCHARGE_EN_OVRD	Override value for BB_LDO_VCOLO_FASTCHARGE_EN When BB_LDO_VCOLO_FASTCHARGE_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_vcolo_fastcharge_en". This bit is ignored when BB_LDO_VCOLO_FASTCHARGE_EN_OVRD_EN==0.
18	Override control for BB_LDO_VCOLO_FASTCHARGE_EN 0b - Normal operation.

Table continues on the next page...

Field	Function
BB_LDO_VCOLO_FASTCHARGE_EN_OVRD_EN	1b - Use the state of BB_LDO_VCOLO_FASTCHARGE_EN_OVRD to override the signal "bb_ldo_vcolo_fastcharge_en".
17 BB_LDO_VCOLO_BLEED_EN_OVRD	Override value for BB_LDO_VCOLO_BLEED_EN When BB_LDO_VCOLO_BLEED_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_vcolo_bleed_en". This bit is ignored when BB_LDO_VCOLO_BLEED_EN_OVRD_EN==0.
16 BB_LDO_VCOLO_BLEED_EN_OVRD_EN	Override control for BB_LDO_VCOLO_BLEED_EN 0b - Normal operation. 1b - Use the state of BB_LDO_VCOLO_BLEED_EN_OVRD to override the signal "bb_ldo_vcolo_bleed_en".
15 BB_LDO_FDBK_BLEED_EN_OVRD	Override value for BB_LDO_FDBK_BLEED_EN When BB_LDO_FDBK_BLEED_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_fdbk_bleed_en". This bit is ignored when BB_LDO_FDBK_BLEED_EN_OVRD_EN==0.
14 BB_LDO_FDBK_BLEED_EN_OVRD_EN	Override control for BB_LDO_FDBK_BLEED_EN 0b - Normal operation. 1b - Use the state of BB_LDO_FDBK_BLEED_EN_OVRD to override the signal "bb_ldo_fdbk_bleed_en".
13 BB_LDO_VTREF_EN_OVRD	Override value for BB_LDO_VTREF_EN When BB_LDO_VTREF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_vtref_en". This bit is ignored when BB_LDO_VTREF_EN_OVRD_EN==0.
12 BB_LDO_VTREF_EN_OVRD_EN	Override control for BB_LDO_VTREF_EN 0b - Normal operation. 1b - Use the state of BB_LDO_VTREF_EN_OVRD to override the signal "bb_ldo_vtref_en".
11 BB_LDO_VCOLO_EN_OVRD	Override value for BB_LDO_VCOLO_EN When BB_LDO_VCOLO_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_vcolo_en". This bit is ignored when BB_LDO_VCOLO_EN_OVRD_EN==0.
10 BB_LDO_VCOLO_EN_OVRD_EN	Override control for BB_LDO_VCOLO_EN 0b - Normal operation. 1b - Use the state of BB_LDO_VCOLO_EN_OVRD to override the signal "bb_ldo_vcolo_en".
9 BB_LDO_FDBK_EN_OVRD	Override value for BB_LDO_FDBK_EN When BB_LDO_FDBK_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_fdbk_en". This bit is ignored when BB_LDO_FDBK_EN_OVRD_EN==0.
8 BB_LDO_FDBK_EN_OVRD_EN	Override control for BB_LDO_FDBK_EN 0b - Normal operation. 1b - Use the state of BB_LDO_FDBK_EN_OVRD to override the signal "bb_ldo_fdbk_en".
7 BB_LDO_PD_EN_OVRD	Override value for BB_LDO_PD_EN When BB_LDO_PD_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_pd_en". This bit is ignored when BB_LDO_PD_EN_OVRD_EN==0.
6 BB_LDO_PD_EN_OVRD_EN	Override control for BB_LDO_PD_EN 0b - Normal operation. 1b - Use the state of BB_LDO_PD_EN_OVRD to override the signal "bb_ldo_pd_en".

Table continues on the next page...

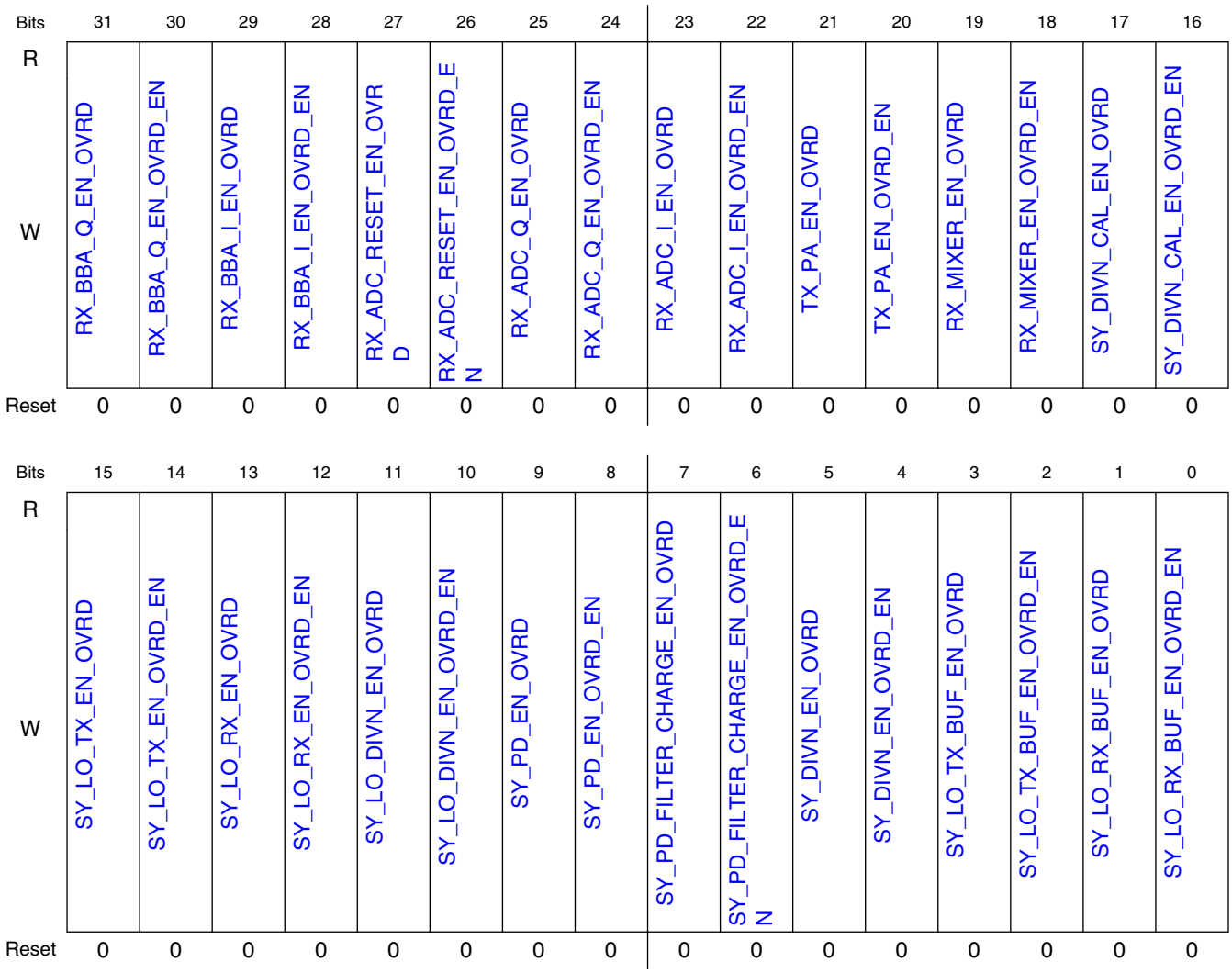
Field	Function
5 BB_LDO_BBA_EN_OVRD	Override value for BB_LDO_BBA_EN When BB_LDO_BBA_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_bba_en". This bit is ignored when BB_LDO_BBA_EN_OVRD_EN==0.
4 BB_LDO_BBA_EN_OVRD_EN	Override control for BB_LDO_BBA_EN 0b - Normal operation. 1b - Use the state of BB_LDO_BBA_EN_OVRD to override the signal "bb_ldo_bba_en".
3 BB_LDO_ADCDAC_EN_OVRD	Override value for BB_LDO_ADCDAC_EN When BB_LDO_ADCDAC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_adcdac_en". This bit is ignored when BB_LDO_ADCDAC_EN_OVRD_EN==0.
2 BB_LDO_ADCDAC_EN_OVRD_EN	Override control for BB_LDO_ADCDAC_EN 0b - Normal operation. 1b - Use the state of BB_LDO_ADCDAC_EN_OVRD to override the signal "bb_ldo_adcdac_en".
1 BB_LDO_HF_EN_OVRD	Override value for BB_LDO_HF_EN When BB_LDO_HF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "bb_ldo_hf_en". This bit is ignored when BB_LDO_HF_EN_OVRD_EN==0.
0 BB_LDO_HF_EN_OVRD_EN	Override control for BB_LDO_HF_EN 0b - Normal operation. 1b - Use the state of BB_LDO_HF_EN_OVRD to override the signal "bb_ldo_hf_en".

A.2.4.72 TSM OVERRIDE REGISTER 1 (OVRD1)

A.2.4.72.1 Offset

Register	Offset
OVRD1	120h

A.2.4.72.2 Diagram



A.2.4.72.3 Fields

Field	Function
31 RX_BBA_Q_EN_OVRD	Override value for RX_BBA_Q_EN When RX_BBA_Q_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_bba_q_en". This bit is ignored when RX_BBA_Q_EN_OVRD_EN==0.
30 RX_BBA_Q_EN_OVRD_EN	Override control for RX_BBA_Q_EN 0b - Normal operation. 1b - Use the state of RX_BBA_Q_EN_OVRD to override the signal "rx_bba_q_en".
29 RX_BBA_I_EN_OVRD	Override value for RX_BBA_I_EN When RX_BBA_I_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_bba_i_en". This bit is ignored when RX_BBA_I_EN_OVRD_EN==0.
28	Override control for RX_BBA_I_EN

Table continues on the next page...

Field	Function
RX_BBA_I_EN_OVRD_EN	0b - Normal operation. 1b - Use the state of RX_BBA_I_EN_OVRD to override the signal "rx_bba_i_en".
27 RX_ADC_RESET_EN_OVRD	Override value for RX_ADC_RESET_EN When RX_ADC_RESET_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_adc_reset_en". This bit is ignored when RX_ADC_RESET_EN_OVRD_EN==0.
26 RX_ADC_RESET_EN_OVRD_EN	Override control for RX_ADC_RESET_EN 0b - Normal operation. 1b - Use the state of RX_ADC_RESET_EN_OVRD to override the signal "rx_adc_reset_en".
25 RX_ADC_Q_EN_OVRD	Override value for RX_ADC_Q_EN When RX_ADC_Q_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_adc_q_en". This bit is ignored when RX_ADC_Q_EN_OVRD_EN==0.
24 RX_ADC_Q_EN_OVRD_EN	Override control for RX_ADC_Q_EN 0b - Normal operation. 1b - Use the state of RX_ADC_Q_EN_OVRD to override the signal "rx_adc_q_en".
23 RX_ADC_I_EN_OVRD	Override value for RX_ADC_I_EN When RX_ADC_I_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_adc_i_en". This bit is ignored when RX_ADC_I_EN_OVRD_EN==0.
22 RX_ADC_I_EN_OVRD_EN	Override control for RX_ADC_I_EN 0b - Normal operation. 1b - Use the state of RX_ADC_I_EN_OVRD to override the signal "rx_adc_i_en".
21 TX_PA_EN_OVRD	Override value for TX_PA_EN When TX_PA_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tx_pa_en". This bit is ignored when TX_PA_EN_OVRD_EN==0.
20 TX_PA_EN_OVRD_EN	Override control for TX_PA_EN 0b - Normal operation. 1b - Use the state of TX_PA_EN_OVRD to override the signal "tx_pa_en".
19 RX_MIXER_EN_OVRD	Override value for RX_MIXER_EN When RX_MIXER_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_mixer_en". This bit is ignored when RX_MIXER_EN_OVRD_EN==0.
18 RX_MIXER_EN_OVRD_EN	Override control for RX_MIXER_EN 0b - Normal operation. 1b - Use the state of RX_MIXER_EN_OVRD to override the signal "rx_mixer_en".
17 SY_DIVN_CAL_EN_OVRD	Override value for SY_DIVN_CAL_EN When SY_DIVN_CAL_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_divn_cal_en". This bit is ignored when SY_DIVN_CAL_EN_OVRD_EN==0.
16 SY_DIVN_CAL_EN_OVRD_EN	Override control for SY_DIVN_CAL_EN 0b - Normal operation. 1b - Use the state of SY_DIVN_CAL_EN_OVRD to override the signal "sy_divn_cal_en".
15 SY_LO_TX_EN_OVRD	Override value for SY_LO_TX_EN When SY_LO_TX_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_lo_tx_en". This bit is ignored when SY_LO_TX_EN_OVRD_EN==0.
14 SY_LO_TX_EN_OVRD_EN	Override control for SY_LO_TX_EN 0b - Normal operation. 1b - Use the state of SY_LO_TX_EN_OVRD to override the signal "sy_lo_tx_en".

Table continues on the next page...

Field	Function
13 SY_LO_RX_EN_OVRD	Override value for SY_LO_RX_EN When SY_LO_RX_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_lo_rx_en". This bit is ignored when SY_LO_RX_EN_OVRD_EN==0.
12 SY_LO_RX_EN_OVRD_EN	Override control for SY_LO_RX_EN 0b - Normal operation. 1b - Use the state of SY_LO_RX_EN_OVRD to override the signal "sy_lo_rx_en".
11 SY_LO_DIVN_EN_OVRD	Override value for SY_LO_DIVN_EN When SY_LO_DIVN_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_lo_divn_en". This bit is ignored when SY_LO_DIVN_EN_OVRD_EN==0.
10 SY_LO_DIVN_EN_OVRD_EN	Override control for SY_LO_DIVN_EN 0b - Normal operation. 1b - Use the state of SY_LO_DIVN_EN_OVRD to override the signal "sy_lo_divn_en".
9 SY_PD_EN_OVRD	Override value for SY_PD_EN When SY_PD_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_pd_en". This bit is ignored when SY_PD_EN_OVRD_EN==0.
8 SY_PD_EN_OVRD_EN	Override control for SY_PD_EN 0b - Normal operation. 1b - Use the state of SY_PD_EN_OVRD to override the signal "sy_pd_en".
7 SY_PD_FILTER_CHARGE_EN_OVRD	Override value for SY_PD_FILTER_CHARGE_EN When SY_PD_FILTER_CHARGE_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_pd_filter_charge_en". This bit is ignored when SY_PD_FILTER_CHARGE_EN_OVRD_EN==0.
6 SY_PD_FILTER_CHARGE_EN_OVRD_EN	Override control for SY_PD_FILTER_CHARGE_EN 0b - Normal operation. 1b - Use the state of SY_PD_FILTER_CHARGE_EN_OVRD to override the signal "sy_pd_filter_charge_en".
5 SY_DIVN_EN_OVRD	Override value for SY_DIVN_EN When SY_DIVN_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_divn_en". This bit is ignored when SY_DIVN_EN_OVRD_EN==0.
4 SY_DIVN_EN_OVRD_EN	Override control for SY_DIVN_EN 0b - Normal operation. 1b - Use the state of SY_DIVN_EN_OVRD to override the signal "sy_divn_en".
3 SY_LO_TX_BUF_EN_OVRD	Override value for SY_LO_TX_BUF_EN When SY_LO_TX_BUF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_lo_tx_buf_en". This bit is ignored when SY_LO_TX_BUF_EN_OVRD_EN==0.
2 SY_LO_TX_BUF_EN_OVRD_EN	Override control for SY_LO_TX_BUF_EN 0b - Normal operation. 1b - Use the state of SY_LO_TX_BUF_EN_OVRD to override the signal "sy_lo_tx_buf_en".
1 SY_LO_RX_BUF_EN_OVRD	Override value for SY_LO_RX_BUF_EN When SY_LO_RX_BUF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "sy_lo_rx_buf_en". This bit is ignored when SY_LO_RX_BUF_EN_OVRD_EN==0.
0 SY_LO_RX_BUF_EN_OVRD_EN	Override control for SY_LO_RX_BUF_EN 0b - Normal operation. 1b - Use the state of SY_LO_RX_BUF_EN_OVRD to override the signal "sy_lo_rx_buf_en".

A.2.4.73 TSM OVERRIDE REGISTER 2 (OVRD2)

A.2.4.73.1 Offset

Register	Offset
OVRD2	124h

A.2.4.73.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	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	RX_PHY_EN_OVRD	RX_PHY_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	TX_DIG_EN_OVRD	TX_DIG_EN_OVRD_EN
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	PLL_DIG_EN_OVRD	PLL_DIG_EN_OVRD_EN	RX_TZA_DCOC_EN_OVRD	RX_TZA_DCOC_EN_OVRD_EN	RX_TZA_PDET_EN_OVRD	RX_TZA_PDET_EN_OVRD_EN	RX_TZA_Q_EN_OVRD	RX_TZA_Q_EN_OVRD_EN	RX_TZA_I_EN_OVRD	RX_TZA_I_EN_OVRD_EN	RX_LNA_EN_OVRD	RX_LNA_EN_OVRD_EN	RX_BBA_DCOC_EN_OVRD	RX_BBA_DCOC_EN_OVRD_EN	RX_BBA_PDET_EN_OVRD	RX_BBA_PDET_EN_OVRD_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.4.73.3 Fields

Field	Function
31	Override value for FREQ_TARG_LD_EN

Table continues on the next page...

Field	Function
FREQ_TARG_LD_EN_OVRD	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.
30 FREQ_TARG_LD_EN_OVRD_EN	Override control for FREQ_TARG_LD_EN 0b - Normal operation. 1b - Use the state of FREQ_TARG_LD_EN_OVRD to override the signal "freq_targ_ld_en".
29 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.
28 DCOC_INIT_OVRD_EN	Override control for DCOC_INIT 0b - Normal operation. 1b - Use the state of DCOC_INIT_OVRD to override the signal "dcoc_init".
27 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.
26 DCOC_EN_OVRD_EN	Override control for DCOC_EN 0b - Normal operation. 1b - Use the state of DCOC_EN_OVRD to override the signal "dcoc_en".
25 RX_PHY_EN_OVRD	Override value for RX_PHY_EN When RX_PHY_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_phy_en". This bit is ignored when RX_PHY_EN_OVRD_EN==0.
24 RX_PHY_EN_OVRD_EN	Override control for RX_PHY_EN 0b - Normal operation. 1b - Use the state of RX_PHY_EN_OVRD to override the signal "rx_phy_en".
23 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.
22 SIGMA_DELTA_EN_OVRD_EN	Override control for SIGMA_DELTA_EN 0b - Normal operation. 1b - Use the state of SIGMA_DELTA_EN_OVRD to override the signal "sigma_delta_en".
21 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.
20 RX_INIT_OVRD_EN	Override control for RX_INIT 0b - Normal operation. 1b - Use the state of RX_INIT_OVRD to override the signal "rx_init".
19 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.
18 RX_DIG_EN_OVRD_EN	Override control for RX_DIG_EN 0b - Normal operation. 1b - Use the state of RX_DIG_EN_OVRD to override the signal "rx_dig_en".
17 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.

Table continues on the next page...

Field	Function
16 TX_DIG_EN_OVRD_EN	Override control for TX_DIG_EN 0b - Normal operation. 1b - Use the state of TX_DIG_EN_OVRD to override the signal "tx_dig_en".
15 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.
14 PLL_DIG_EN_OVRD_EN	Override control for PLL_DIG_EN 0b - Normal operation. 1b - Use the state of PLL_DIG_EN_OVRD to override the signal "pll_dig_en".
13 RX_TZA_DCOC_EN_OVRD	Override control for RX_TZA_DCOC_EN 0b - Normal operation. 1b - Use the state of RX_TZA_DCOC_EN_OVRD to override the signal "rx_tza_dcoc_en".
12 RX_TZA_DCOC_EN_OVRD_EN	Override control for RX_TZA_DCOC_EN 0b - Normal operation. 1b - Use the state of RX_TZA_DCOC_EN_OVRD to override the signal "rx_tza_dcoc_en".
11 RX_TZA_PDET_EN_OVRD	Override value for RX_TZA_PDET_EN When RX_TZA_PDET_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_tza_pdet_en". This bit is ignored when RX_TZA_PDET_EN_OVRD_EN==0.
10 RX_TZA_PDET_EN_OVRD_EN	Override control for RX_TZA_PDET_EN 0b - Normal operation. 1b - Use the state of RX_TZA_PDET_EN_OVRD to override the signal "rx_tza_pdet_en".
9 RX_TZA_Q_EN_OVRD	Override value for RX_TZA_Q_EN When RX_TZA_Q_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_tza_q_en". This bit is ignored when RX_TZA_Q_EN_OVRD_EN==0.
8 RX_TZA_Q_EN_OVRD_EN	Override control for RX_TZA_Q_EN 0b - Normal operation. 1b - Use the state of RX_TZA_Q_EN_OVRD to override the signal "rx_tza_q_en".
7 RX_TZA_I_EN_OVRD	Override value for RX_TZA_I_EN When RX_TZA_I_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_tza_i_en". This bit is ignored when RX_TZA_I_EN_OVRD_EN==0.
6 RX_TZA_I_EN_OVRD_EN	Override control for RX_TZA_I_EN 0b - Normal operation. 1b - Use the state of RX_TZA_I_EN_OVRD to override the signal "rx_tza_i_en".
5 RX_LNA_EN_OVRD	Override value for RX_LNA_EN When RX_LNA_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_lna_en". This bit is ignored when RX_LNA_EN_OVRD_EN==0.
4 RX_LNA_EN_OVRD_EN	Override control for RX_LNA_EN 0b - Normal operation. 1b - Use the state of RX_LNA_EN_OVRD to override the signal "rx_lna_en".
3 RX_BBA_DCOC_EN_OVRD	Override value for RX_BBA_DCOC_EN When RX_BBA_DCOC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_bba_dcoc_en". This bit is ignored when RX_BBA_DCOC_EN_OVRD_EN==0.
2	Override control for RX_BBA_DCOC_EN 0b - Normal operation.

Table continues on the next page...

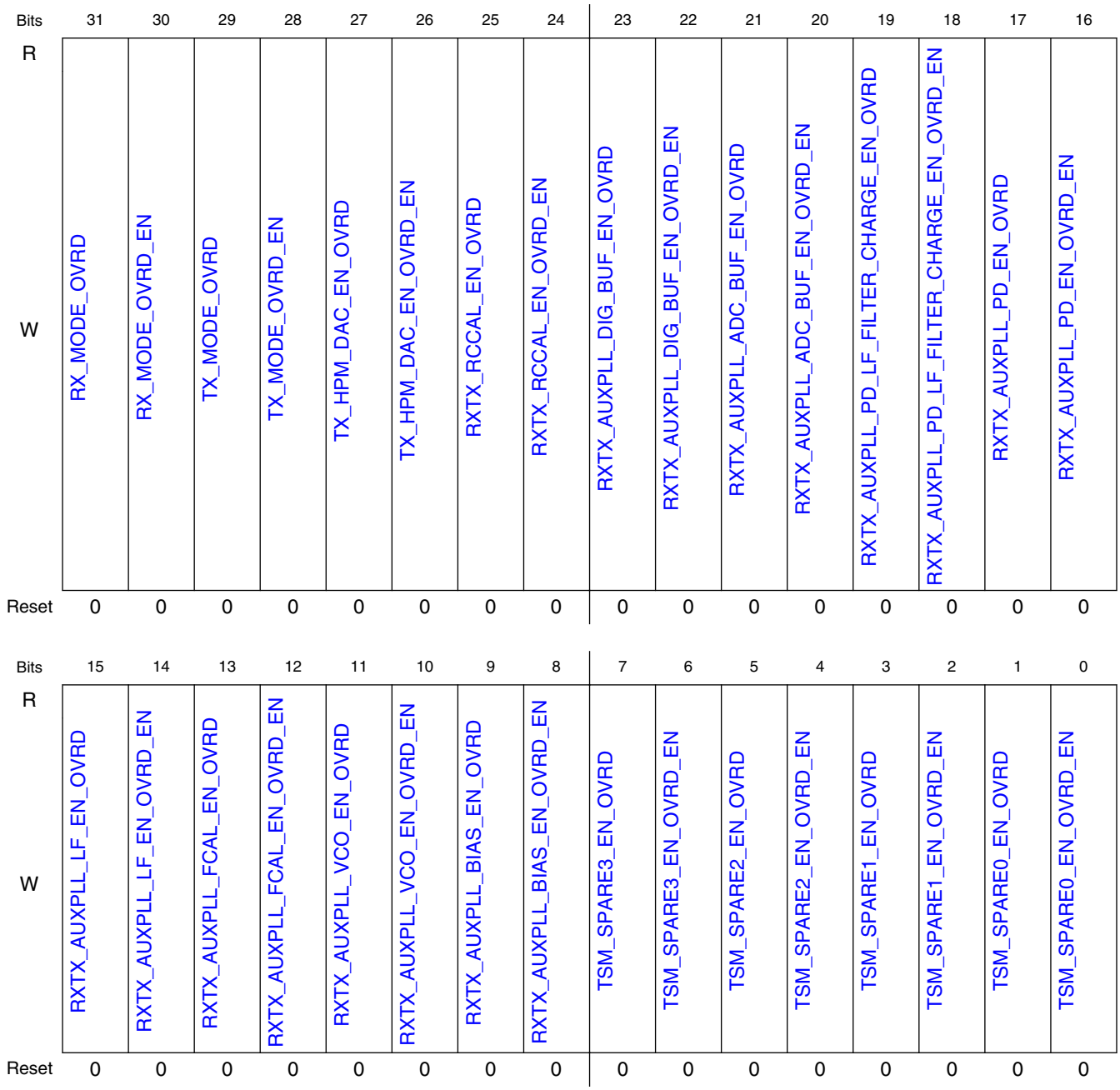
Field	Function
RX_BBA_DCOC_EN_OVRD_EN	1b - Use the state of RX_BBA_DCOC_EN_OVRD to override the signal "rx_bba_dcoc_en".
1 RX_BBA_PDET_EN_OVRD	Override value for RX_BBA_PDET_EN When RX_BBA_PDET_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rx_bba_pdet_en". This bit is ignored when RX_BBA_PDET_EN_OVRD_EN==0.
0 RX_BBA_PDET_EN_OVRD_EN	Override control for RX_BBA_PDET_EN 0b - Normal operation. 1b - Use the state of RX_BBA_PDET_EN_OVRD to override the signal "rx_bba_pdet_en".

A.2.4.74 TSM OVERRIDE REGISTER 3 (OVRD3)

A.2.4.74.1 Offset

Register	Offset
OVRD3	128h

A.2.4.74.2 Diagram



A.2.4.74.3 Fields

Field	Function
31 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.

Table continues on the next page...

Field	Function
30 RX_MODE_OVRD_EN	Override control for RX_MODE 0b - Normal operation. 1b - Use the state of RX_MODE_OVRD to override the signal "rx_mode".
29 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.
28 TX_MODE_OVRD_EN	Override control for TX_MODE 0b - Normal operation. 1b - Use the state of TX_MODE_OVRD to override the signal "tx_mode".
27 TX_HPM_DAC_EN_OVRD	Override value for TX_HPM_DAC_EN When TX_HPM_DAC_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "tx_hpm_dac_en". This bit is ignored when TX_HPM_DAC_EN_OVRD_EN==0.
26 TX_HPM_DAC_EN_OVRD_EN	Override control for TX_HPM_DAC_EN 0b - Normal operation. 1b - Use the state of TX_HPM_DAC_EN_OVRD to override the signal "tx_hpm_dac_en".
25 RXTX_RCCAL_EN_OVRD	Override value for RXTX_RCCAL_EN When RXTX_RCCAL_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_rccal_en". This bit is ignored when RXTX_RCCAL_EN_OVRD_EN==0.
24 RXTX_RCCAL_EN_OVRD_EN	Override control for RXTX_RCCAL_EN 0b - Normal operation. 1b - Use the state of RXTX_RCCAL_EN_OVRD to override the signal "rxtx_rccal_en".
23 RXTX_AUXPLL_DIG_BUF_EN_OVRD	Override value for RXTX_AUXPLL_DIG_BUF_EN When RXTX_AUXPLL_DIG_BUF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_dig_buf_en". This bit is ignored when RXTX_AUXPLL_DIG_BUF_EN_OVRD_EN==0.
22 RXTX_AUXPLL_DIG_BUF_EN_OVRD_EN	Override control for RXTX_AUXPLL_DIG_BUF_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_DIG_BUF_EN_OVRD to override the signal "rxtx_auxpll_dig_buf_en".
21 RXTX_AUXPLL_ADC_BUF_EN_OVRD	Override value for RXTX_AUXPLL_ADC_BUF_EN When RXTX_AUXPLL_ADC_BUF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_adc_buf_en". This bit is ignored when RXTX_AUXPLL_ADC_BUF_EN_OVRD_EN==0.
20 RXTX_AUXPLL_ADC_BUF_EN_OVRD_EN	Override control for RXTX_AUXPLL_ADC_BUF_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_ADC_BUF_EN_OVRD to override the signal "rxtx_auxpll_adc_buf_en".
19 RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_OVRD	Override value for RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN When RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_pd_lf_filter_charge_en". This bit is ignored when RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_OVRD_EN==0.
18	Override control for RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_OVRD to override the signal "rxtx_auxpll_pd_lf_filter_charge_en".

Table continues on the next page...

Field	Function
RXTX_AUXPLL_PD_LF_FILTER_CHARGE_EN_OVRD_EN	
17 RXTX_AUXPLL_PD_EN_OVRD	Override value for RXTX_AUXPLL_PD_EN When RXTX_AUXPLL_PD_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_pd_en". This bit is ignored when RXTX_AUXPLL_PD_EN_OVRD_EN==0.
16 RXTX_AUXPLL_PD_EN_OVRD_EN	Override control for RXTX_AUXPLL_PD_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_PD_EN_OVRD to override the signal "rxtx_auxpll_pd_en".
15 RXTX_AUXPLL_LF_EN_OVRD	Override value for RXTX_AUXPLL_LF_EN When RXTX_AUXPLL_LF_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_lf_en". This bit is ignored when RXTX_AUXPLL_LF_EN_OVRD_EN==0.
14 RXTX_AUXPLL_LF_EN_OVRD_EN	Override control for RXTX_AUXPLL_LF_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_LF_EN_OVRD to override the signal "rxtx_auxpll_lf_en".
13 RXTX_AUXPLL_FCAL_EN_OVRD	Override value for RXTX_AUXPLL_FCAL_EN When RXTX_AUXPLL_FCAL_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_fcal_en". This bit is ignored when RXTX_AUXPLL_FCAL_EN_OVRD_EN==0.
12 RXTX_AUXPLL_FCAL_EN_OVRD_EN	Override control for RXTX_AUXPLL_FCAL_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_FCAL_EN_OVRD to override the signal "rxtx_auxpll_fcal_en".
11 RXTX_AUXPLL_VCO_EN_OVRD	Override value for RXTX_AUXPLL_VCO_EN When RXTX_AUXPLL_VCO_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_vco_en". This bit is ignored when RXTX_AUXPLL_VCO_EN_OVRD_EN==0.
10 RXTX_AUXPLL_VCO_EN_OVRD_EN	Override control for RXTX_AUXPLL_VCO_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_VCO_EN_OVRD to override the signal "rxtx_auxpll_vco_en".
9 RXTX_AUXPLL_BIAS_EN_OVRD	Override value for RXTX_AUXPLL_BIAS_EN When RXTX_AUXPLL_BIAS_EN_OVRD_EN=1, this value overrides the mission mode state of the signal "rxtx_auxpll_bias_en". This bit is ignored when RXTX_AUXPLL_BIAS_EN_OVRD_EN==0.
8 RXTX_AUXPLL_BIAS_EN_OVRD_EN	Override control for RXTX_AUXPLL_BIAS_EN 0b - Normal operation. 1b - Use the state of RXTX_AUXPLL_BIAS_EN_OVRD to override the signal "rxtx_auxpll_bias_en".
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	Override control for TSM_SPARE3_EN 0b - Normal operation.

Table continues on the next page...

Field	Function
TSM_SPARE3_EN_OVRD_EN	1b - 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 0b - Normal operation. 1b - 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 0b - Normal operation. 1b - 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 0b - Normal operation. 1b - Use the state of TSM_SPARE0_EN_OVRD to override the signal "tsm_spare0_en".

A.2.5 XCVR_MISC register descriptions

A.2.5.1 XCVR_CTRL_ADDR Memory map

XCVR_MISC base address: 4005_C280h

Offset	Register	Width (In bits)	Access	Reset value
0h	TRANSCEIVER CONTROL (XCVR_CTRL)	32	RW	0010_1000h
4h	TRANSCEIVER STATUS (XCVR_STATUS)	32	W1C	See description.
8h	BLE ARBITRATION CONTROL (BLE_ARB_CTRL)	32	RW	0000_0000h
Ch	OVERWRITE VERSION (OVERWRITE_VER)	32	RW	0000_0000h
10h	DIGITAL TEST MUX CONTROL (DTEST_CTRL)	32	RW	0000_0000h
14h	TRANSCEIVER DMA CONTROL (DMA_CTRL)	32	RW	0000_0300h
18h	TRANSCEIVER DMA DATA (DMA_DATA)	32	RO	0000_0000h
1Ch	PACKET RAM CONTROL (PACKET_RAM_CTRL)	32	RW	0000_0000h

Table continues on the next page...

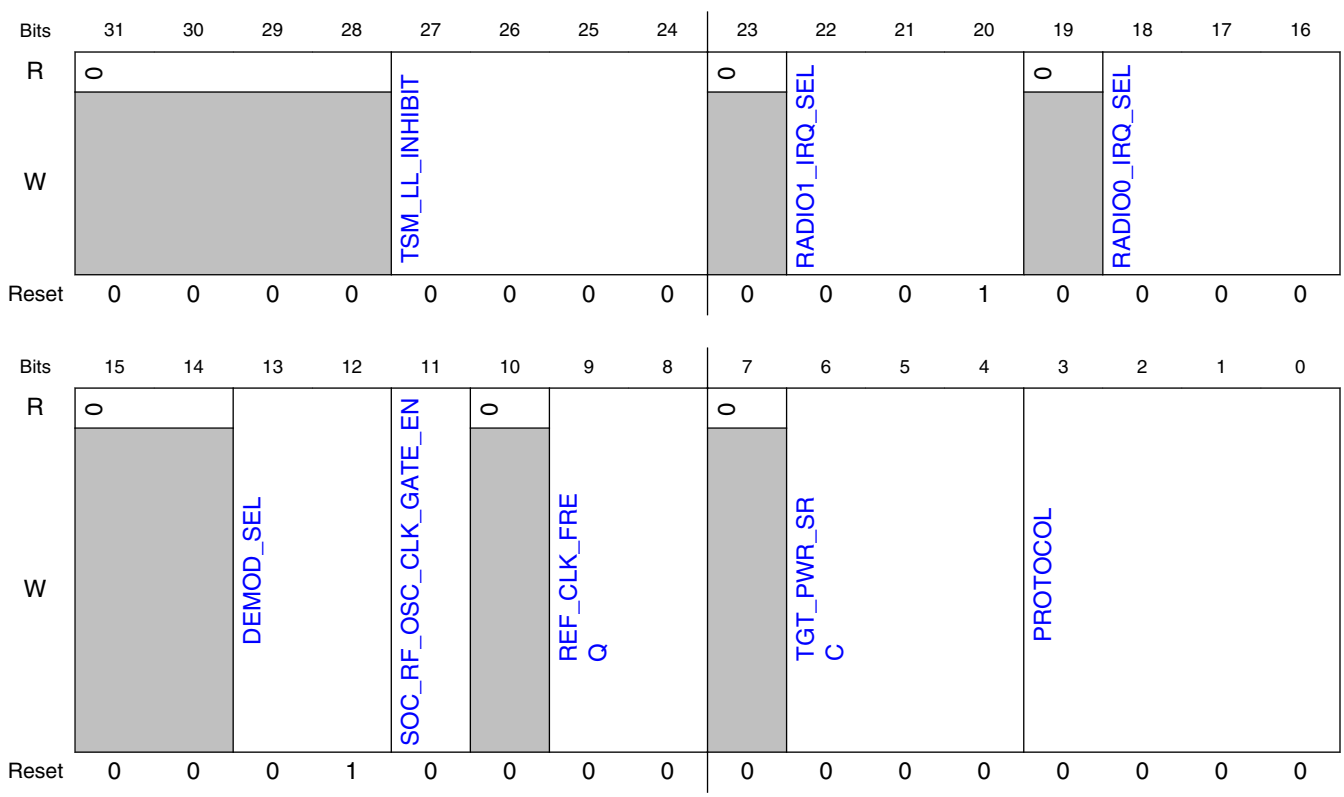
Offset	Register	Width (In bits)	Access	Reset value
20h	PACKET RAM DEBUG RAM STOP ADDRESS (RAM_STOP_ADDR)	32	RO	0000_0000h
24h	FAD CONTROL (FAD_CTRL)	32	RW	0000_F080h
2Ch	COEXISTENCE CONTROL (COEX_CTRL)	32	RW	0000_0000h
30h	CRC/WHITENER CONFIG REGISTER (CRCW_CFG)	32	RW	0000_0000h
34h	CRC ERROR CORRECTION MASK (CRC_EC_MASK)	32	RO	0000_0000h
38h	CRC RESULT (CRC_RES_OUT)	32	RO	0000_0000h
3Ch	CRC/WHITENER CONFIG 2 REGISTER (CRCW_CFG2)	32	RW	0000_0000h

A.2.5.2 TRANSCEIVER CONTROL (XCVR_CTRL)

A.2.5.2.1 Offset

Register	Offset
XCVR_CTRL	0h

A.2.5.2.2 Diagram



A.2.5.2.3 Fields

Field	Function
31-28 —	Reserved
27-24 TSM_LL_INHIBIT	TSM Per-Link-Layer Inhibit Nominally, any protocol engine can request TSM to start a TX or RX sequence at any time. In a multiprotocol setting, individual protocol engines can (optionally) be selectively blocked from accessing TSM by setting one of the following bits of TSM_LL_INHIBIT[3:0]: **xxx1**: BLE commands to TSM are inhibited **xxx0**: BLE commands to TSM are permitted **1xxx**: GENERIC_FSK commands to TSM are inhibited **0xxx**: GENERIC_FSK commands to TSM are permitted **NOTE:** TSM_LL_INHIBIT[1] and TSM_LL_INHIBIT[2] currently have no functionality
23 —	Reserved
22-20 RADIO1_IRQ_SEL	RADIO1_IRQ_SEL Assigns Radio #1 Interrupt (ipi_int_radio1) to a Protocol Engine 000b - Assign Radio #1 Interrupt to BLE 001b - Radio #1 Interrupt unassigned 010b - Radio #1 Interrupt unassigned 011b - Assign Radio #1 Interrupt to GENERIC_FSK 100b - Radio #1 Interrupt unassigned 101b - Radio #1 Interrupt unassigned 110b - Radio #1 Interrupt unassigned 111b - Radio #1 Interrupt unassigned
19 —	Reserved
18-16 RADIO0_IRQ_SEL	RADIO0_IRQ_SEL Assigns Radio #0 Interrupt (ipi_int_radio0) to a Protocol Engine 000b - Assign Radio #0 Interrupt to BLE 001b - Radio #0 Interrupt unassigned 010b - Radio #0 Interrupt unassigned 011b - Assign Radio #0 Interrupt to GENERIC_FSK 100b - Radio #0 Interrupt unassigned 101b - Radio #0 Interrupt unassigned 110b - Radio #0 Interrupt unassigned 111b - Radio #0 Interrupt unassigned
15-14 —	Reserved.
13-12 DEMOD_SEL	Demodulator Selector This bit selects the demodulator used during reception 00b - No demodulator selected 01b - Use NXP Multi-standard PHY demodulator 10b - Reserved 11b - Reserved

Table continues on the next page...

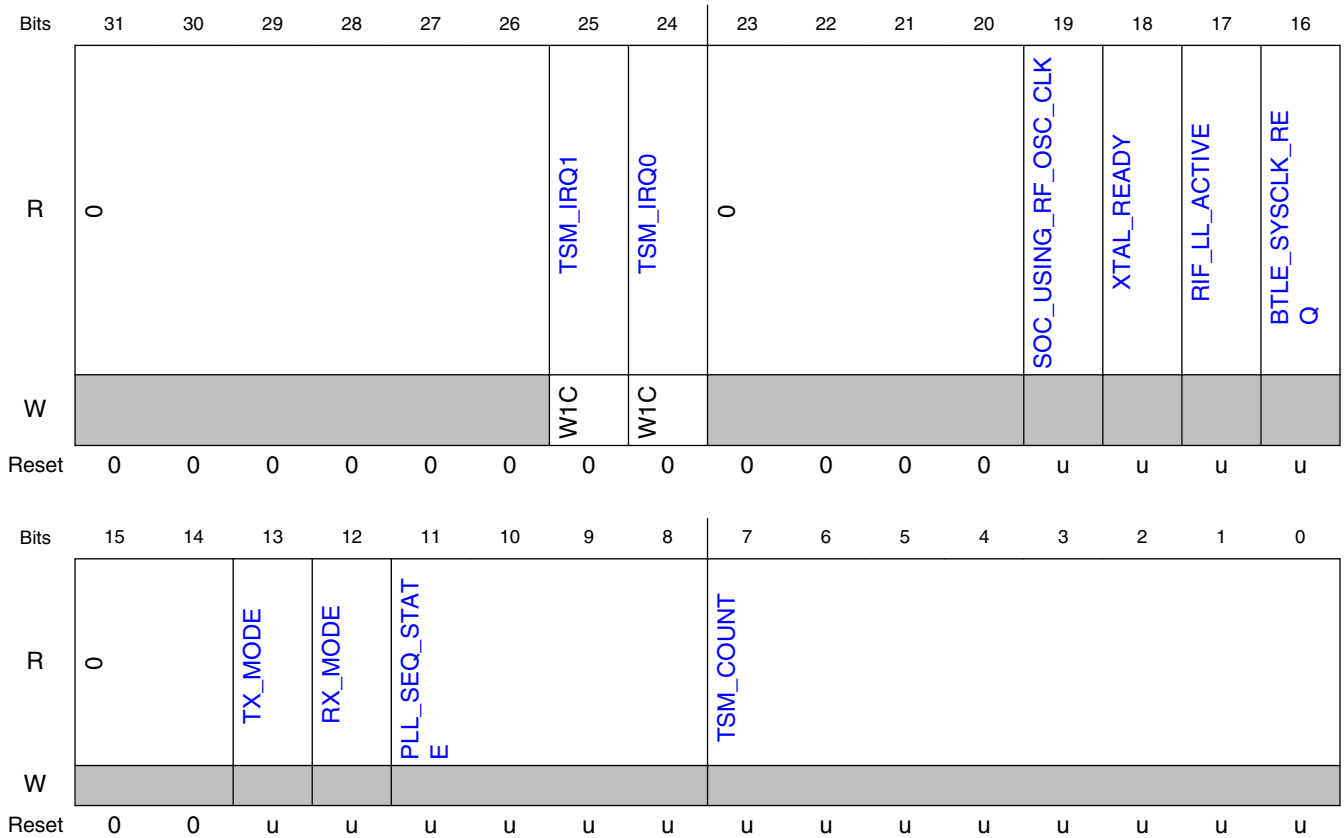
Field	Function
11 SOC_RF_OSC_CLK_GATE_EN	SOC_RF_OSC_CLK_GATE_EN Enable 3V version of RF OSC Clock for use by the SoC
10 —	Reserved
9-8 REF_CLK_FREQ	Radio Reference Clock Frequency This register selects the Reference Clock Frequency for the Radio. 00b - 32 MHz 01b - 26 MHz 10b - Reserved 11b - Reserved
7 —	Reserved
6-4 TGT_PWR_SRC	Target Power Source For determining transmit power, the TGT_PWR_SRC[2:0] bits control target power selection, according to the following table. 0: TARGET POWER SOURCE is PA_POWER[5:0] register (XCVR space) 1: TARGET POWER SOURCE is BTLE Link Layer 2: Reserved 3: Reserved 4: TARGET POWER SOURCE is GENERIC_FSK Link Layer (TX_POWER[5:0] register in GENERIC_FSK space) 5: Reserved 6: Reserved 7: TARGET POWER SOURCE is determined by the PROTOCOL[3:0] bits
3-0 PROTOCOL	Radio Protocol Selection This register selects the Radio Communication Protocol. 0000b - BLE 0001b - BLE in MBAN 0010b - BLE overlap MBAN 0011b - Reserved 0100b - Reserved 0101b - Reserved 0110b - Radio Channels 0-127 selectable, FSK 0111b - Radio Channels 0-127 selectable, GFSK 1000b - Generic GFSK, with Gaussian Filter 1001b - Generic MSK, O-QPSK encoding 1010b - Generic FSK, direct +/- Fdev FSK

A.2.5.3 TRANSCEIVER STATUS (XCVR_STATUS)

A.2.5.3.1 Offset

Register	Offset
XCVR_STATUS	4h

A.2.5.3.2 Diagram



A.2.5.3.3 Fields

Field	Function
31-26 —	Reserved
25 TSM_IRQ1	TSM Interrupt #1 0b - TSM Interrupt #1 is not asserted. 1b - TSM Interrupt #1 is asserted. Write '1' to this bit to clear it.
24 TSM_IRQ0	TSM Interrupt #0 0b - TSM Interrupt #0 is not asserted. 1b - TSM Interrupt #0 is asserted. Write '1' to this bit to clear it.
23-20 —	Reserved
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	RF Osciallator Xtal Ready

Table continues on the next page...

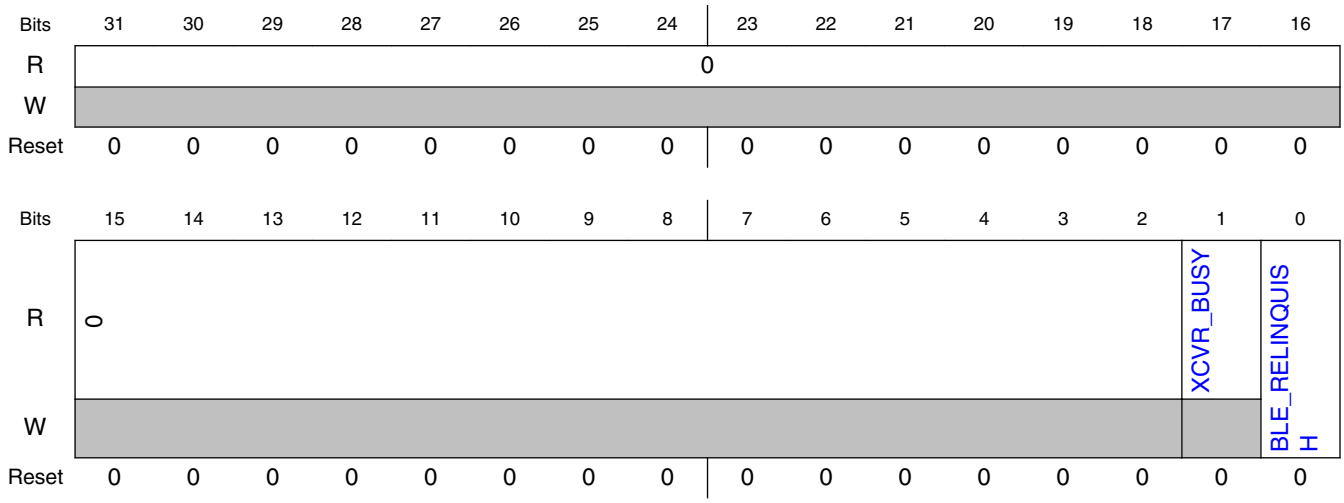
Field	Function
XTAL_READY	Oscillator warmup count complete. 0b - Indicates that the RF Oscillator is disabled or has not completed its warmup. 1b - Indicates that the RF Oscillator has completed its warmup count and is ready for use.
17 RIF_LL_ACTIVE	Link Layer Active Indication Reflects the state of 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.
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
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. 0000b - PLL OFF 0010b - CTUNE 0011b - CTUNE_SETTLE 0110b - HPMCAL1 1000b - HPMCAL1_SETTLE 1010b - HPMCAL2 1100b - HPMCAL2_SETTLE 1111b - PLLREADY
7-0 TSM_COUNT	TSM_COUNT Reflects the instantaneous value of the TSM counter.

A.2.5.4 BLE ARBITRATION CONTROL (BLE_ARB_CTRL)

A.2.5.4.1 Offset

Register	Offset
BLE_ARB_CTRL	8h

A.2.5.4.2 Diagram



A.2.5.4.3 Fields

Field	Function
31-2 —	Reserved
1 XCVR_BUSY	Transceiver Busy Status Bit 0b - RF Channel in available (TSM is idle) 1b - RF Channel in use (TSM is busy)
0 BLE_RELINQUISH	BLE Relinquish Control This bit forces the BLE protocol engine to immediately relinquish access to the RF Channel, in favor of another protocol with higher arbitration priority. BLE is denied RF channel access by blocking BLE access to the TSM; when this bit is subsequently cleared, BLE access is restored.

A.2.5.5 OVERWRITE VERSION (OVERWRITE_VER)

A.2.5.5.1 Offset

Register	Offset
OVERWRITE_VER	Ch

A.2.5.5.2 Function

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.

A.2.5.5.3 Diagram

Bits	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

Bits	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

A.2.5.5.4 Fields

Field	Function
31-8 —	Reserved
7-0 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.

A.2.5.6 DIGITAL TEST MUX CONTROL (DTEST_CTRL)

A.2.5.6.1 Offset

Register	Offset
DTEST_CTRL	10h

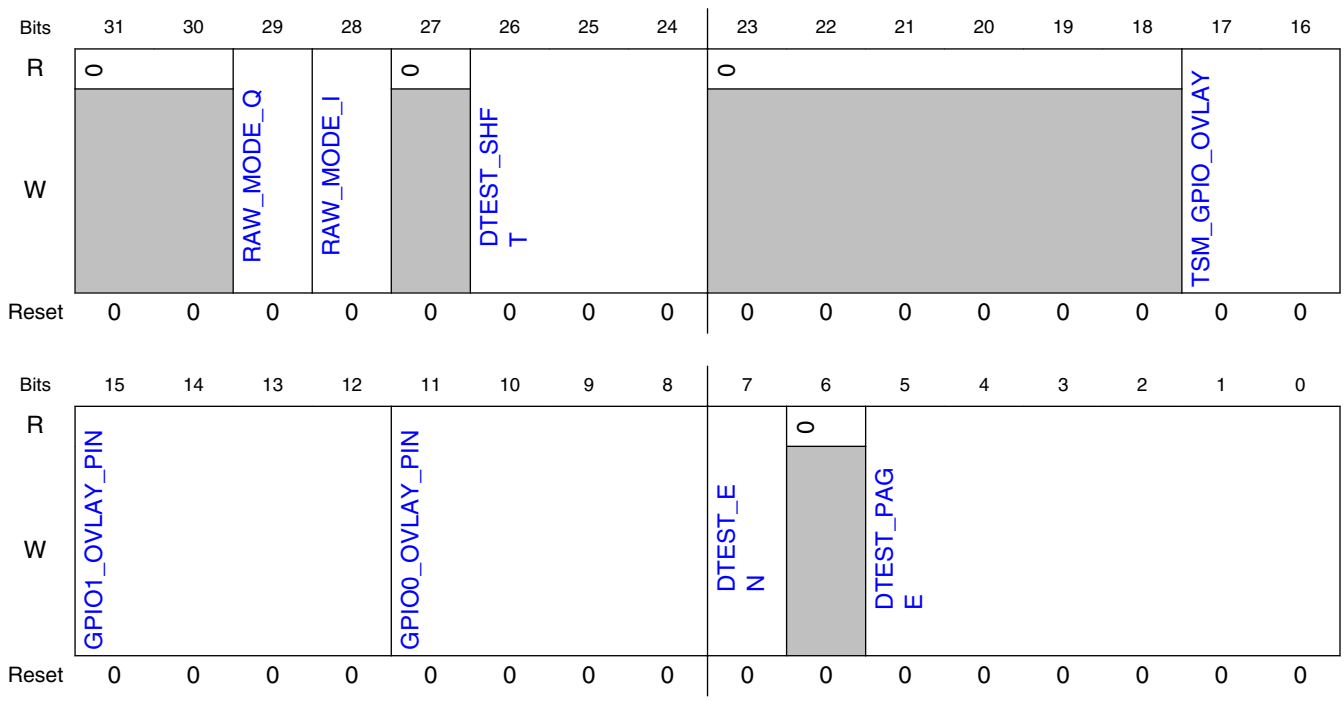
A.2.5.6.2 Function

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

A.2.5.6.3 Diagram



A.2.5.6.4 Fields

Field	Function
31-30 —	Reserved
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

Table continues on the next page...

Field	Function
	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.
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: PLL_RIPPLE_COUNTER[18:0] 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[18:0], an unsigned value, are shifted by DTEST_SHFT[2:0] according to the following table 0: PLL_RIPPLE_COUNTER[18:5] appear on DTEST[13:0] 1: PLL_RIPPLE_COUNTER[17:4] appear on DTEST[13:0] 2: PLL_RIPPLE_COUNTER[16:3] appear on DTEST[13:0] 3: PLL_RIPPLE_COUNTER[15:2] appear on DTEST[13:0] 4: PLL_RIPPLE_COUNTER[14:1] appear on DTEST[13:0] 5: PLL_RIPPLE_COUNTER[13:0] appear on DTEST[13:0] 6: Reserved 7: Reserved 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: 0: These bits appear on DTEST[13:2] : RX_DIG_IQ[11:0] (no shift) 1: These bits appear on DTEST[13:2] : RX_DIG_IQ[11],RX_DIG_IQ[11:1] (right shift by 1) 2: These bits appear on DTEST[13:2] : RX_DIG_IQ[11],RX_DIG_IQ[11],RX_DIG_IQ[11:2] (right shift by 2) 3: These bits appear on DTEST[13:2] : RX_DIG_IQ[11:0] (no shift)
23-18 —	Reserved
17-16 TSM_GPIO_OVLAY	TSM GPIO Overlay Pin Control 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. Similarly, the TSM-controlled output GPIO1_TRIG_EN can be routed to any DTEST pin, 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

Table continues on the next page...

Field	Function
	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[0] = 0 and TSM_GPIO_OVLAY[1] = 0, there is no overlay, and the DTEST Page Table dictates the node that appears on each DTEST pin. 00b - there is no overlay, and the DTEST Page Table dictates the node that appears on each DTEST pin. 01b - the register GPIO0_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO0_TRIG_EN will appear. 10b - the register GPIO1_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO1_TRIG_EN will appear. 11b - the register GPIO0_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO0_TRIG_EN will appear, and the register GPIO1_OVLAY_PIN[3:0] selects the DTEST pin on which GPIO1_TRIG_EN will appear.
15-12 GPIO1_OVLAY_PIN	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, 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 0b - Disables DTEST. The DTEST pins assume their mission function. 1b - Enables DTEST. The contents of the selected page (DTEST_PAGE) will appear on the DTEST output pins.
6 —	Reserved
5-0 DTEST_PAGE	DTEST Page Selector DTEST Page signal assignments 000000b - PLLFREQCAL 000001b - PLLBESTDIFF 000010b - PLLRIPPLE 000011b - PLLHPMCAL 000100b - PLLVCOMOD 000101b - PLLUNLOCK 000110b - PLLCYCSLIP 000111b - PLLCHAN 001000b - TXWARMUP 001001b - TXPOWER 001010b - TXFREQWORD 001011b - RXWARMUP 001100b - RXADC 001101b - RXDMA 001110b - RXDIGIQ 001111b - RXDMA2 010000b - RXINPH 010001b - RSSI0 010010b - RSSI1

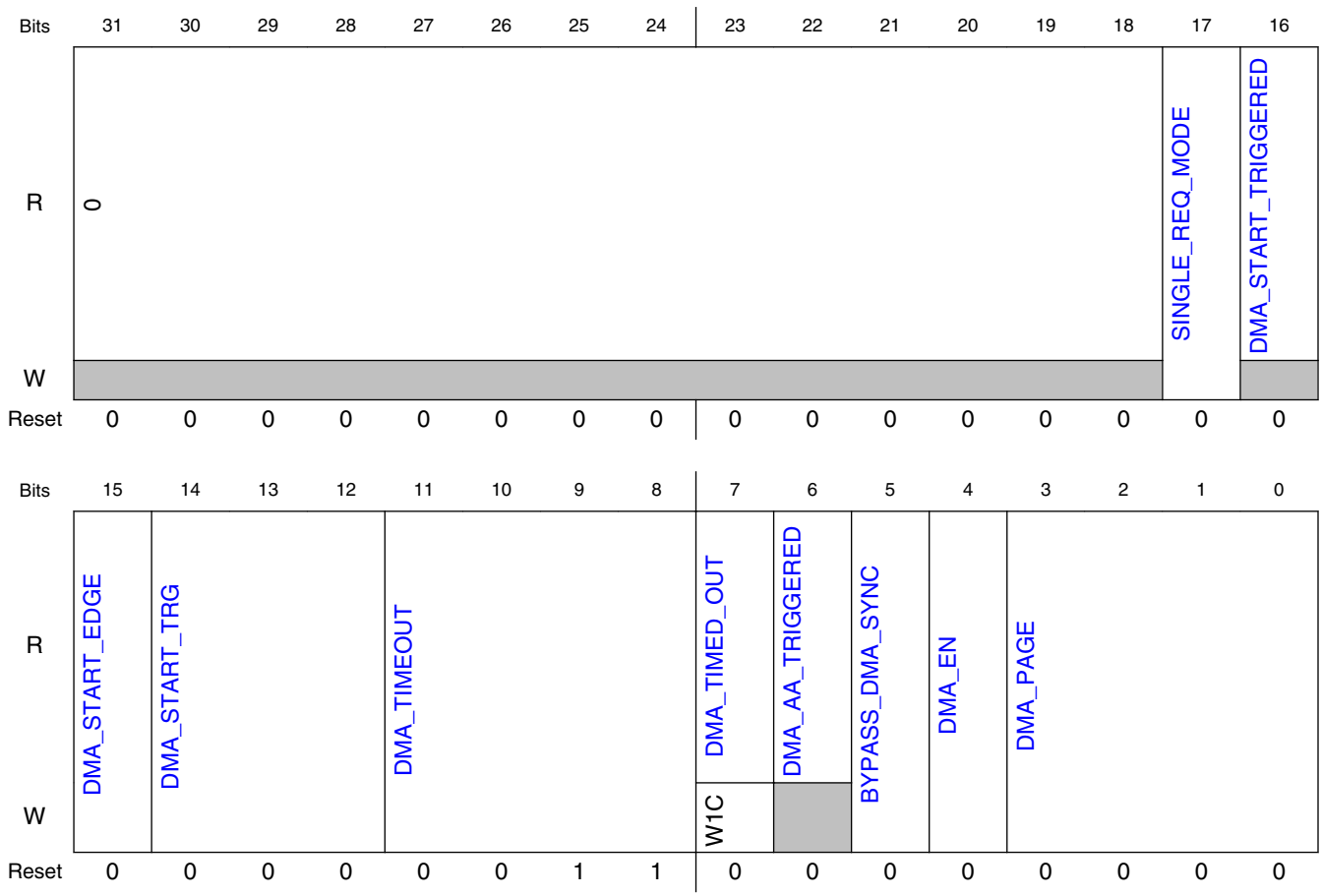
Field	Function
	010011b - AGC0 010100b - AGC1 010101b - DCOC0 010110b - DCOC1 010111b - DCOC2 011000b - DCOC3 011001b - TSM 011010b - MTTSMCAL 011011b - MTADV 011100b - MTINIT 011101b - MTSCAN 011110b - MTCONN 011111b - MTDTM 100000b - MTADVXCV 100001b - MTCONXCV 100010b - MTDTM2 100011b - DSM 100100b - PHY_FSK_STATE 100101b - PHY_CFO_EST_PD 100110b - PHY_CFO_EST_PD2 100111b - PHY_EARLY_LATE 101000b - PHY_FSK_DEMOD 101001b - PHY_AA_SEARCH 101010b - PHY_DATA_OUT 101011b - PHY_SAMP_TIME 101100b - CCA_ED_LQI 101101b - CCA_ED_LQI2 101110b - Reserved 101111b - Reserved 110000b - Reserved 110001b - Reserved 110010b - Reserved 110011b - Reserved 110100b - Reserved 110101b - Reserved 110110b - Reserved 110111b - Reserved 111000b - Reserved 111001b - Reserved 111010b - RCCAL 111011b - AUXPLLFCAL 111100b - GENFSKTX 111101b - GENFSKRX 111110b - GENFSKSTATE 111111b - GENFILTER

A.2.5.7 TRANSCEIVER DMA CONTROL (DMA_CTRL)

A.2.5.7.1 Offset

Register	Offset
DMA_CTRL	14h

A.2.5.7.2 Diagram



A.2.5.7.3 Fields

Field	Function
31-18 —	Reserved
17 SINGLE_REQ_MODE	DMA Single Request Mode Configures the transceiver to transfer the entire block of DMA data with only a single initial request (ipd_req_radio_rx). The DMA2 controller must be configured accordingly. In this mode, the transceiver will use **ips_xfr_wait** to pace the individual transactions. Single Request Mode should not be used with

Table continues on the next page...

Field	Function
	DMA Pages 11, 12, or 13, because the data rate is too low and therefore ips_xfr_wait would remain assserted for excessively long periods. 0b - Disable Single Request Mode. The transceiver will assert ipd_req_radio_rx whenever it has a new sample ready for transfer. 1b - Enable Single Request Mode. A single initial request by the transceiver will transfer the entire DMA block of data
16 DMA_START_TRIGGERED	DMA Start Trigger Occurred This read-only status bit becomes set, when, during a DMA session (DMA_PAGE > 0), the trigger source selected by DMA_START_TRG[2:0] occurs, with the edge sensitivity selected by DMA_START_EDGE. To clear this bit, set DMA_PAGE=0.
15 DMA_START_EDGE	DMA Start Trigger Edge Selector DMA_START_EDGE selects the edge sensitivity (rising or falling) of the selected start-trigger. 0b - Trigger fires on a rising edge of the selected trigger source 1b - Trigger fires on a falling edge of the selected trigger source
14-12 DMA_START_TRG	DMA Start Trigger Selector The DMA Start Trigger, if desired, can be selected from the sources in the following table: 0: no trigger 1: FSK PHY: gfsk_preamble_found 2: FSK PHY: aa_sfd_matched 3: Reserved 4: Reserved 5: RXDIG: agc_dcoc_gain_chg 6: TSM: rx_dig_en 7: TSM: tsm_spare2_en
11-8 DMA_TIMEOUT	DMA Timeout In DMA Single Request Mode, adverse consequences may result if the transceiver's ips_xfr_wait signal is asserted for an excessively long period of time, which could occur due to mis-programming of the DMA2 controller, and/or the transceiver itself. DMA Timeout forces the transceiver's ips_xfr_wait low after *N* microseconds and sets the **DMA_TIMED_OUT** status bit, where *N*=DMA_TIMEOUT. DMA_TIMEOUT is only relavent when SINGLE_REQ_MODE=1, otherwise the transceiver does not assert ips_xfr_wait.
7 DMA_TIMED_OUT	DMA Transfer Timed Out Status bit indicates that the transceiver DMA, while operating in Single Request Mode, asserted ips_xfr_wait for a period in excess of the programmed **DMA_TIMEOUT** setting, resulting in a timeout condition where the transceiver has forced ips_xfr_wait low. After a timeout, before resuming DMA operations, set DMA_PAGE=0 to disable DMA, and write 1 to this bit to clear it. 0b - A DMA timeout has not occurred 1b - A DMA timeout has occurred in Single Request Mode since the last time this bit was cleared
6 DMA_AA_TRIGGERED	DMA Access Address triggered This status bit becomes set when the PHY indicates that a Network Address match has occurred. In some cases it may be desirable to delay the start of DMA transfers until such a match occurs; in such cases, software can poll this bit, and once it becomes set, immediately select the desired DMA_PAGE to initiate DMA transfers. This bit is intended for use with DMA_PAGE=13, but is available on all pages. Its usage is optional.
5 BYPASS_DMA_SYNC	Bypass External DMA Synchronization When using transceiver DMA with SINGLE_REQ_MODE=0, synchronization external to the transceiver must be bypassed in order to minimize bus access latency. Using DMA in this mode also requires that the MCU and transceiver both are configured to operate in the RF Oscillator clock domain. 0b - Don't Bypass External Synchronization. Use this setting if SINGLE_REQ_MODE=1. 1b - Bypass External Synchronization. This setting is mandatory if SINGLE_REQ_MODE=0.
4	DMA Enable

Table continues on the next page...

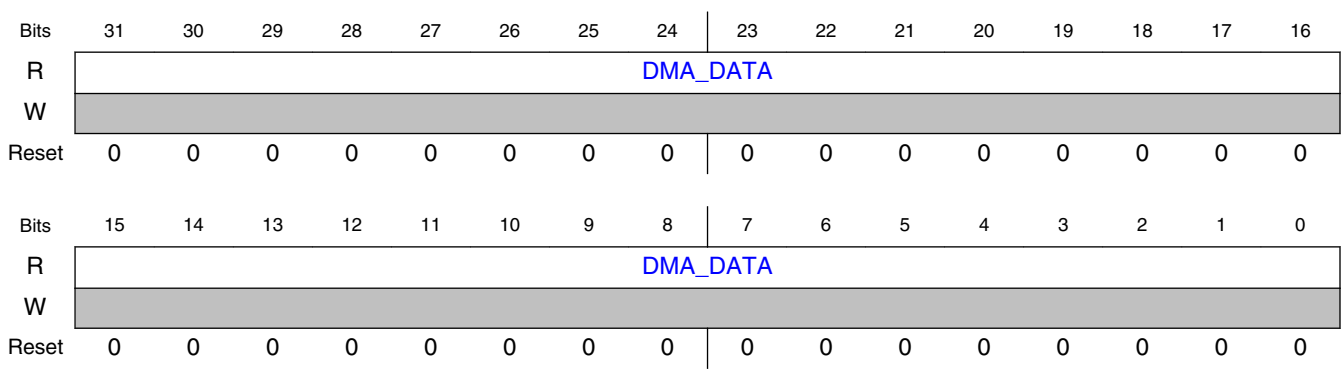
Field	Function
DMA_EN	Setting DMA_EN=1 enables clock gating to the Transciever DMA engine. This bit should be set prior to the start of a DMA session, and remain set for the duration of the session.
3-0 DMA_PAGE	Transceiver DMA Page Selector Selects the page of receiver data for storage to system memory when using Transceiver DMA Debug Mode. Setting this register to a non-zero value enables the DMA interface logic. Setting this register to zero disables the interface. **Note:** DMA_PAGE should be set to a non-zero value to begin an acquisition, only after the start trigger has been programmed (DMA_START_TRG), if a trigger is desired. The available DMA pages are listed below. 0000b - DMA Idle 0001b - RX_DIG I and Q 0010b - RX_DIG I Only 0011b - RX_DIG Q Only 0100b - RAW ADC I and Q 0101b - RAW ADC I Only 0110b - RAW ADC Q only 0111b - DC Estimator I and Q 1000b - DC Estimator I Only 1001b - DC Estimator Q only 1010b - RX_DIG Phase Output 1011b - Reserved 1100b - Demodulator Soft Decision 1101b - Demodulator Data Output 1110b - Demodulator CFO Phase Output 1111b - Reserved

A.2.5.8 TRANSCEIVER DMA DATA (DMA_DATA)

A.2.5.8.1 Offset

Register	Offset
DMA_DATA	18h

A.2.5.8.2 Diagram



A.2.5.8.3 Fields

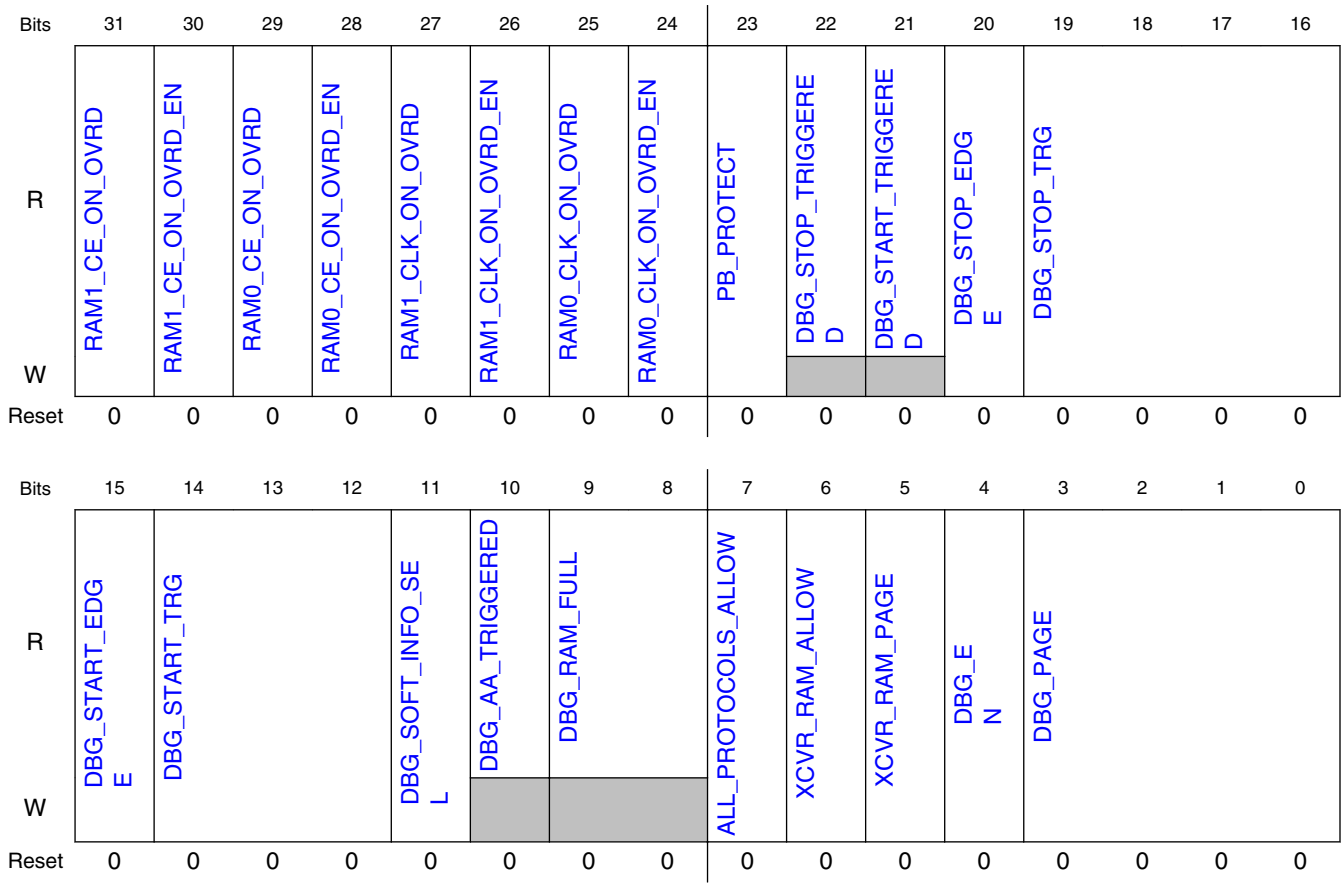
Field	Function
31-0 DMA_DATA	DMA Data Register This register is the singular address location at which the SoC DMA controller accesses samples from the transceiver for transfer to system memory. During DMA operation, the contents of this register depend on the DMA Page selected. The DMA Data register is intended for DMA purposes only, not Host IPS accesses, but Host IPS bus access to this register is not prohibited.

A.2.5.9 PACKET RAM CONTROL (PACKET_RAM_CTRL)

A.2.5.9.1 Offset

Register	Offset
PACKET_RAM_CTRL	1Ch

A.2.5.9.2 Diagram



A.2.5.9.3 Fields

Field	Function
31 RAM1_CE_ON_OVRD	Override value for RAM1 CE (Chip Enable) When RAM1_CE_ON_OVRD_EN=1, this value overrides the mission mode state of the RAM1 CE. This bit is ignored when RAM1_CE_ON_OVRD_EN=0.
30 RAM1_CE_ON_OVRD_EN	Override control for RAM1 CE (Chip Enable) 0b - Normal operation. 1b - Use the state of RAM1_CE_ON_OVRD to override the RAM1 CE.
29 RAM0_CE_ON_OVRD	Override value for RAM0 CE (Chip Enable) When RAM0_CE_ON_OVRD_EN=1, this value overrides the mission mode state of the RAM0 CE. This bit is ignored when RAM0_CE_ON_OVRD_EN=0.
28 RAM0_CE_ON_OVRD_EN	Override control for RAM0 CE (Chip Enable) 0b - Normal operation. 1b - Use the state of RAM0_CE_ON_OVRD to override the RAM0 CE.
27 RAM1_CLK_ON_OVRD	Override value for RAM1 Clock Gate Enable When RAM1_CLK_ON_OVRD_EN=1, this value overrides the mission mode state of the RAM1 Clock Gate Enable. This bit is ignored when RAM1_CLK_ON_OVRD_EN=0.

Table continues on the next page...

Field	Function
26 RAM1_CLK_ON_OVRD_EN	Override control for RAM1 Clock Gate Enable 0b - Normal operation. 1b - Use the state of RAM1_CLK_ON_OVRD to override the RAM1 Clock Gate Enable.
25 RAM0_CLK_ON_OVRD	Override value for RAM0 Clock Gate Enable When RAM0_CLK_ON_OVRD_EN=1, this value overrides the mission mode state of the RAM0 Clock Gate Enable. This bit is ignored when RAM0_CLK_ON_OVRD_EN=0.
24 RAM0_CLK_ON_OVRD_EN	Override control for RAM0 Clock Gate Enable 0b - Normal operation. 1b - Use the state of RAM0_CLK_ON_OVRD to override the RAM0 Clock Gate Enable.
23 PB_PROTECT	Packet Buffer Protect Protect Packet Buffer contents against overwriting by the next received packet. Applies to all protocols except BLE 0b - Incoming received packets overwrite Packet Buffer RX contents (default) 1b - Incoming received packets are blocked from overwriting Packet Buffer RX contents
22 DBG_STOP_TRIGGERED	Packet RAM Debug Stop Triggered This read-only status bit becomes set, when, during a Packet RAM Debug session (DBG_PAGE > 0), the trigger source selected by DBG_STOP_TRG[3:0] occurs, with the edge sensitivity selected by DBG_STOP_EDGE. To clear this bit, set DBG_PAGE=0.
21 DBG_START_TRIGGERED	Packet RAM Debug Start Triggered This read-only status bit becomes set, when, during a Packet RAM Debug session (DBG_PAGE > 0), the trigger source selected by DBG_START_TRG[2:0] occurs, with the edge sensitivity selected by DBG_START_EDGE. To clear this bit, set DBG_PAGE=0.
20 DBG_STOP_EDGE	Packet RAM Debug Stop Trigger Edge Selector DBG_STOP_EDGE selects the edge sensitivity (rising or falling) of the selected stop-trigger. 0b - Trigger fires on a rising edge of the selected trigger source 1b - Trigger fires on a falling edge of the selected trigger source
19-16 DBG_STOP_TRG	Packet RAM Debug Stop Trigger Selector The Packet RAM Debug Stop Trigger, if desired, can be selected from the sources in the following table: 0: no trigger 1: FSK PHY: gfsk_preamble_found 2: FSK PHY: aa_sfd_matched 3: Reserved 4: Reserved 5: RXDIG: agc_dcoc_gain_chg 6: TSM: rx_dig_en 7: TSM: tsm_spare3_en 8: TSM: pll unlock 9: BLE: crc_err_count_incr (increment DTM RX bad packet counter) 10: CRC FAIL: GEN_FSK only 11: HEADER FAIL: GEN_FSK only 12-15: Reserved
15 DBG_START_EDGE	Packet RAM Debug Start Trigger Edge Selector DBG_START_EDGE selects the edge sensitivity (rising or falling) of the selected start-trigger. 0b - Trigger fires on a rising edge of the selected trigger source 1b - Trigger fires on a falling edge of the selected trigger source
14-12 DBG_START_TRG	Packet RAM Debug Start Trigger Selector The Packet RAM Debug Start Trigger, if desired, can be selected from the sources in the following table:

Table continues on the next page...

Field	Function
	0: no trigger 1: FSK PHY: gfsk_preamble_found 2: FSK PHY: aa_sfd_matched 3: Reserved 4: Reserved 5: RXDIG: agc_dcoc_gain_chg 6: TSM: rx_dig_en 7: TSM: tsm_spare2_en
11 DBG_SOFT_INFO_SEL	Packet RAM Debug PHY Soft Info Output Selector When acquiring data in Packet RAM Debug Mode with the DBG_PAGE=DMDSOFT page, this bit selects the signal from the PHY used to capture the soft decision data. **NOTE:** When the soft bits are captured without regard to aa_sfd_matched, the bits will be sampled at a higher rate before Access Address detection. Prior to the assertion of aa_sfd_matched, the soft bits are at OSR 8. Following the assertion of aa_sfd_matched plus 2, the soft bits are at OSR 2. 0b - PHY input cg_vbr_en is used to capture soft decision data 1b - PHY output fsk_demod_bit_valid is used to capture soft decision data
10 DBG_AA_TRIGGERED	Packet Ram Debug Access Address triggered This status bit becomes set when the PHY indicates that a Network Address match has occurred. In some cases it may be desirable to delay the start of Packet RAM debug transfers until such a match occurs; in such cases, software can poll this bit, and once it becomes set, immediately select the desired DBG_PAGE to initiate data transfers to Packet RAM. This bit is intended for use with DBG_PAGE=13, but is available on all pages. Its usage is optional.
9-8 DBG_RAM_FULL	DBG_RAM_FULL[1:0] Status Bits indicating that Packet RAM0 (or RAM1) is full, and the Packet RAM Debug engine has attempted to write another word to that RAM. This, and any subsequent write attempts, will not alter the contents of the RAM, once it has reached capacity. To clear the DBG_RAM_FULL[1:0] bits, the DBG_PAGE field must be set to 0 and then set to a non-zero value (which starts a new capture by the Packet RAM Debug engine) That is, after a capture, the DBG_RAM_FULL[1:0] bits only reset to zero upon the start of the subsequent capture, at which time it is safe to start polling these bits again. **NOTE:** If a Debug Stop Trigger is selected (DBG_STOP_TRG > 0), writes to RAM do not stop when DBG_RAM_FULL bit(s) become set; instead, writes continue after an address wrap-around to 0, and will continue indefinitely until either a stop trigger condition occurs, or the debug session is ended by taking DBG_PAGE=0. The DBG_RAM_FULL[1:0] remain a valid indicator that the RAM(s) has reached full status at least once during the debug session. 00b - Neither Packet RAM0 nor RAM1 is full 1xb - Packet RAM1 has been filled to capacity. x1b - Packet RAM0 has been filled to capacity.
7 ALL_PROTOCOLS_ALLOW	Allow IPS bus access to Packet RAM for any protocol at any time. Each supported protocol has an associated IPS bus, with which it can access protocol-specific registers, as well as Packet RAM. Normally IPS bus access to the Packet RAM is restricted to the protocol currently selected by the XCVR_CTRL[PROTOCOL] register, and access attempts by other protocols will result in an assert of **ips_xfr_err** on the offending IPS interface. When ALL_PROTOCOLS_ALLOW=1, these inhibits are removed, and any IPS bus can access Packet RAM at any time without any error signalling. 0b - IPS bus access to Packet RAM is restricted to the protocol engine currently selected by XCVR_CTRL[PROTOCOL]. 1b - All IPS bus access to Packet RAM permitted, regardless of XCVR_CTRL[PROTOCOL] setting
6 XCVR_RAM_ALLOW	Allow Packet RAM Transceiver Access This bit must be set before performing accesses to the Packet RAM using transceiver address space. Transceiver space accesses to Packet RAM are intended for debug purposes only; the individual protocol engines, and the associated IPS busses, have exclusive access to Packet RAM in mission modes. Transceiver space accesses to Packet RAM include direct accesses to RAM at transceiver addresses

Table continues on the next page...

Field	Function
	0x700 - 0xFFF, as well as Packet RAM Debug Mode. When this bit is set, control of RAM clock gating and RAM chip enables are forced on continuously, taking these controls away from the protocol engines. 0b - Protocol Engines, and associated IPS busses, have exclusive access to Packet RAM (mission mode) 1b - Transceiver-space access to Packet RAM, including Packet RAM debug mode, are allowed
5 XCVR_RAM_PAGE	RAM Page Selector for XCVR Access This bit selects which of the 2 Packet RAM blocks is mapped into XCVR address space starting at XCVR_BASE + 0x700. 0b - RAM0 is mapped into XCVR address space, between XCVR_BASE + 0x700, and XCVR_BASE + 0xFFF 1b - RAM1 is mapped into XCVR address space, between XCVR_BASE + 0x700, and XCVR_BASE + 0xFFF
4 DBG_EN	Packet RAM Debug Mode Enable Setting DBG_EN=1 enables clock gating to the Packet RAM Debug engine. This bit should be set prior to the start of a Debug session, and remain set for the duration of the session.
3-0 DBG_PAGE	Packet RAM Debug Page Selector Selects the page of receiver data for storage to Packet RAM using Transceiver Packet RAM Debug Mode. Setting this register to a non-zero value enables the Packet RAM Debug Mode interface logic. Setting this register to zero disables the interface. **Note:** DBG_PAGE should be set to a non-zero value to begin an acquisition, only after the start and stop triggers have been programmed (DBG_START_TRG and DBG_STOP_TRG), if a trigger is desired. The available RAM Debug pages are listed below. 0000b - Packet RAM Debug Mode Idle 0001b - RX_DIG I and Q 0010b - Reserved 0011b - Reserved 0100b - RAW ADC I and Q 0101b - Reserved 0110b - Reserved 0111b - DC Estimator I and Q 1000b - Reserved 1001b - Reserved 1010b - RX_DIG Phase Output 1011b - Reserved 1100b - Demodulator Soft Decision 1101b - Demodulator Data Output 1110b - Demodulator CFO Phase Output 1111b - Reserved

A.2.5.10 PACKET RAM DEBUG RAM STOP ADDRESS (RAM_STOP_ADDR)

A.2.5.10.1 Offset

Register	Offset
RAM_STOP_ADDR	20h

A.2.5.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					RAM1_STOP_ADDR										
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	0					RAM0_STOP_ADDR										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.2.5.10.3 Fields

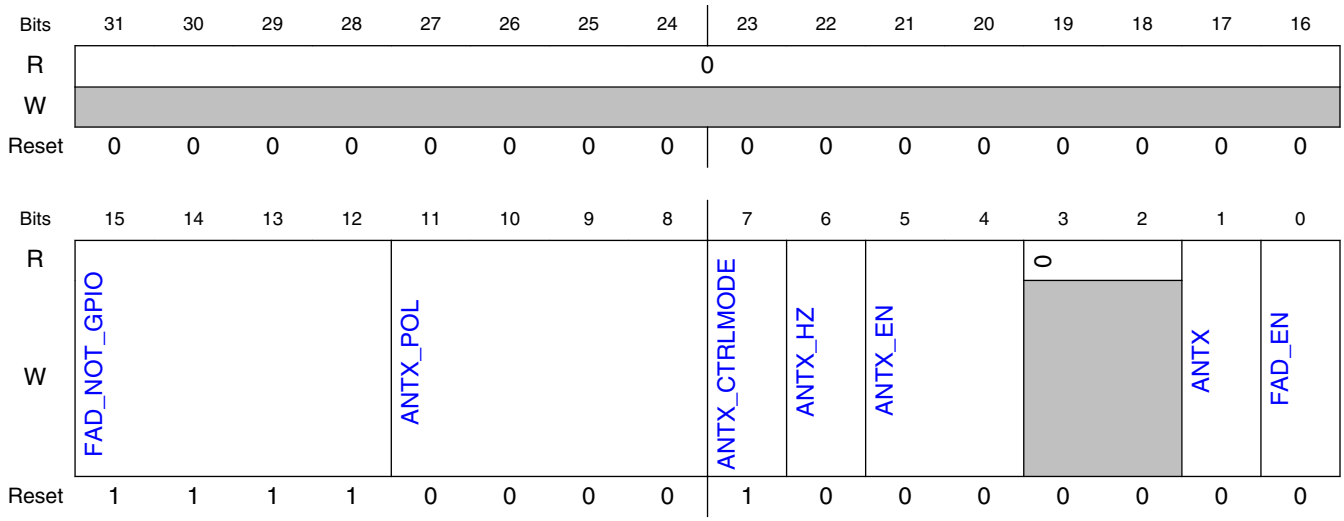
Field	Function
31-27 —	Reserved
26-16 RAM1_STOP_ADDR	RAM1 Stop Address After a Packet RAM Debug Mode acquisition, RAM1_STOP_ADDR represents the last (most recent) RAM1 location filled with sample data before the acquision terminated. The acquisition terminates when a stop-trigger occurs, or due to a manual termination by setting DBG_PAGE=0. RAM1_STOP_ADDR is in units of RAM "words", where each word represents 2 bytes of XCVR RAM address space.
15-11 —	Reserved
10-0 RAM0_STOP_ADDR	RAM0 Stop Address After a Packet RAM Debug Mode acquisition, RAM0_STOP_ADDR represents the last (most recent) RAM0 location filled with sample data before the acquision terminated. The acquisition terminates when a stop-trigger occurs, or due to a manual termination by setting DBG_PAGE=0. RAM0_STOP_ADDR is in units of RAM "words", where each word represents 2 bytes of XCVR RAM address space.

A.2.5.11 FAD CONTROL (FAD_CTRL)

A.2.5.11.1 Offset

Register	Offset
FAD_CTRL	24h

A.2.5.11.2 Diagram



A.2.5.11.3 Fields

Field	Function
31-16 —	Reserved
15-12 FAD_NOT_GPIO	FAD versus GPIO Mode Selector **x1xx**: Reserved **x0xx**: The TX_SWITCH pad is controlled directly by the TSM output gpio2_trig_en **1xxx**: Reserved **0xxx**: The RX_SWITCH pad is controlled directly by the TSM output gpio3_trig_en **NOTE:** FAD_NOT_GPIO[0] and FAD_NOT_GPIO[1] currently have no functionality.
11-8 ANTX_POL	FAD Antenna Controls Polarity Control the polarity of the FAD pins: ANTX_POL<0> : This bit currently has no functionality ANTX_POL<1> : This bit currently has no functionality ANTX_POL<2>=1 : invert the TX_SWITCH output ANTX_POL<3>=1 : invert the RX_SWITCH output
7 ANTX_CTRLMODE	Antenna Diversity Control Mode When ANTX_CTRLMODE=1 (dual mode): TX_SWITCH=GPIO2_TRIG_EN RX_SWITCH=GPIO3_TRIG_EN When ANTX_CTRLMODE=0 (single mode): TX_SWITCH=GPIO2_TRIG_EN RX_SWITCH=(GPIO3_TRIG_EN OR GPIO2_TRIG_EN)

Table continues on the next page...

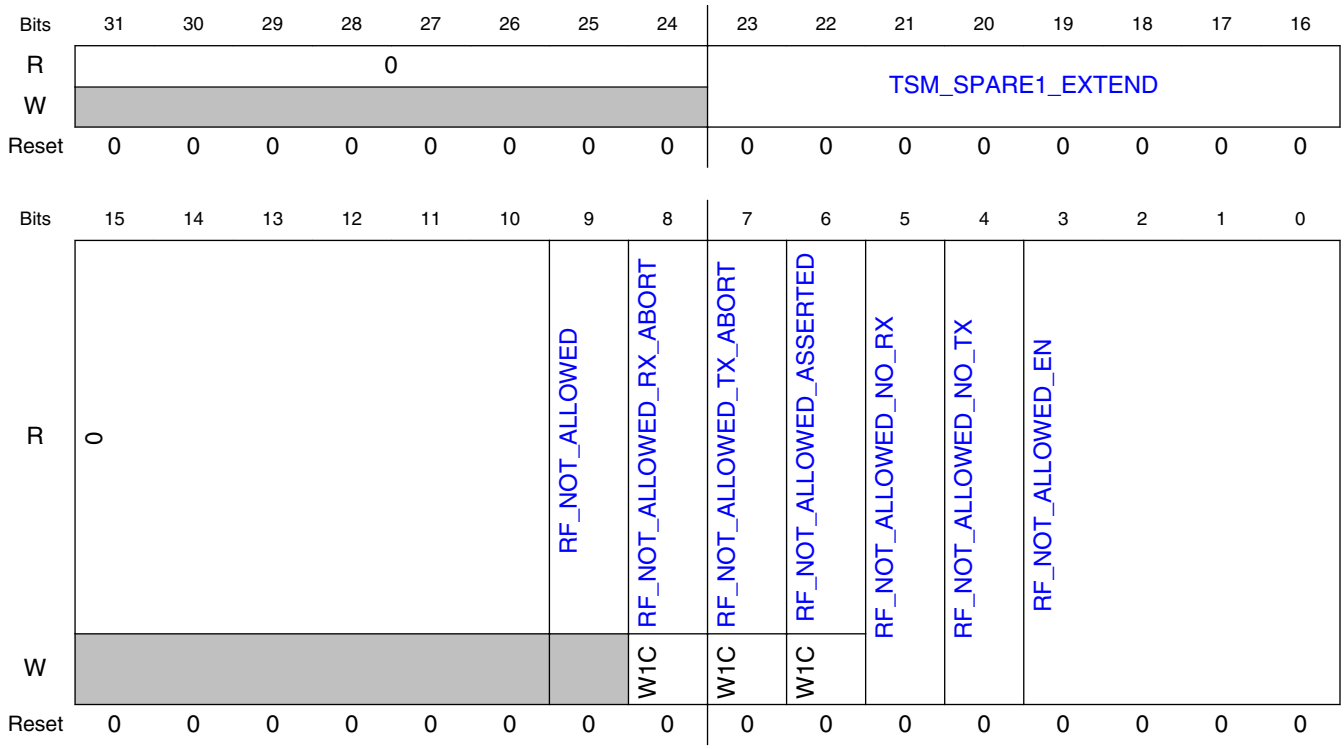
Field	Function
	GPIO2_TRIG_EN and GPIO3_TRIG_EN are outputs of the Transceiver Sequence Manager (TSM). The TSM timing registers associated with GPIO2_TRIG_EN and GPIO3_TRIG_EN should be programmed with the desired TX_SWITCH and RX_SWITCH timing
6 ANTX_HZ	This bit currently has no functionality
5-4 ANTX_EN	These bits currently have no functionality
3-2 —	Reserved.
1 ANTX	This bit currently has no functionality
0 FAD_EN	This bit currently has no functionality

A.2.5.12 COEXISTENCE CONTROL (COEX_CTRL)

A.2.5.12.1 Offset

Register	Offset
COEX_CTRL	2Ch

A.2.5.12.2 Diagram



A.2.5.12.3 Fields

Field	Function
31-24 —	Reserved
23-16 TSM_SPARE1_EXTEND	TSM_SPARE1_EX Extension Duration When TSM output TSM_SPARE1_EN is enabled to assert during any TX or RX sequence and TSM_SPARE1_EXTEND[7:0] > 0, this register specifies the number of microseconds for which TSM_SPARE1_EN is to remain asserted beyond the end of the TSM warmdown. This extension is intended to close any gap in the coexistence output RF_ACTIVE that may occur between consecutive TX/RX or RX/TX TSM operations.
15-10 —	Reserved
9 RF_NOT_ALLOWED	RF_NOT_ALLOWED Reflects the instantaneous state of the RF_NOT_ALLOWED pin, synchronized into the RF OSC clock domain.
8 RF_NOT_ALLOWED_RX_ABORT	RF_NOT_ALLOWED_RX_ABORT 0b - A RX abort due to assertion on RF_NOT_ALLOWED has not occurred 1b - A RX abort due to assertion on RF_NOT_ALLOWED has occurred since the last time this bit was cleared
7	RF_NOT_ALLOWED_TX_ABORT 0b - A TX abort due to assertion on RF_NOT_ALLOWED has not occurred

Table continues on the next page...

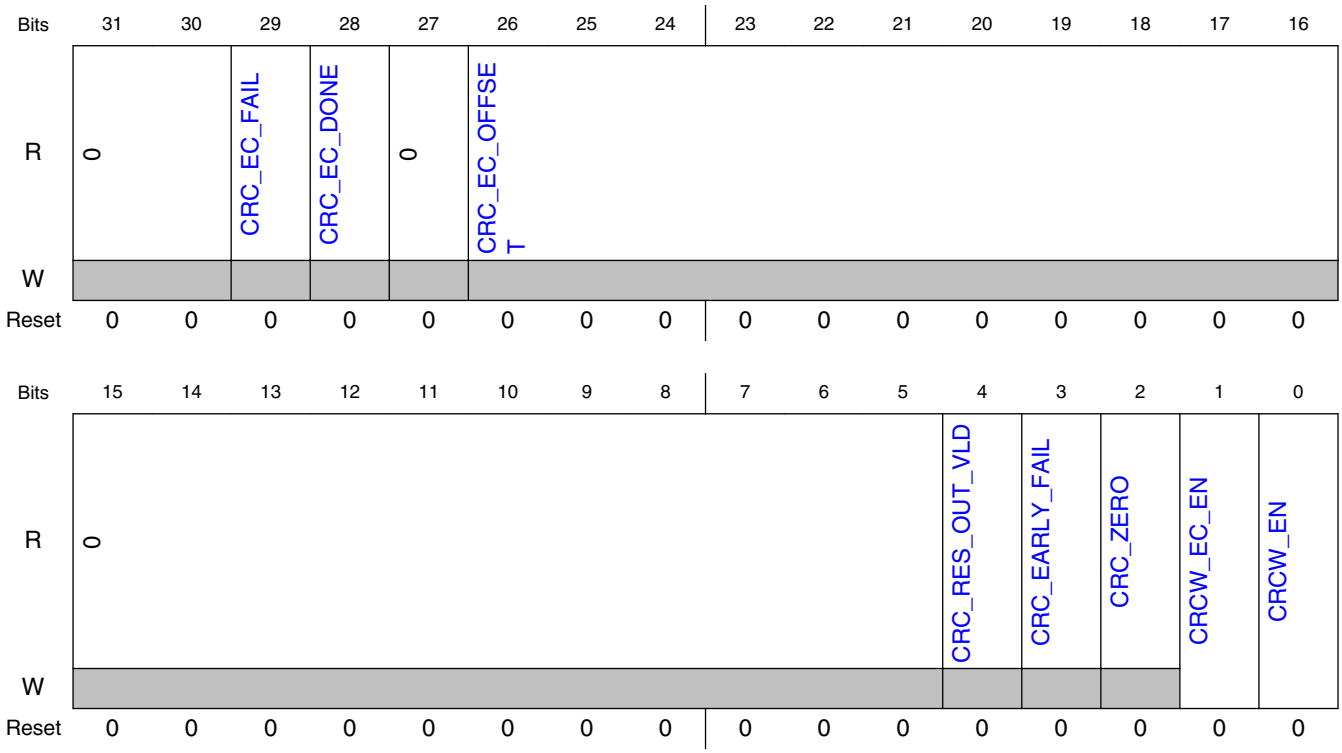
Field	Function
RF_NOT_ALLOWED_TX_ABORT	1b - A TX abort due to assertion on RF_NOT_ALLOWED has occurred since the last time this bit was cleared
6 RF_NOT_ALLOWED_ASSERTED	RF_NOT_ALLOWED_ASSERTED 0b - Assertion on RF_NOT_ALLOWED has not occurred 1b - Assertion on RF_NOT_ALLOWED has occurred since the last time this bit was cleared
5 RF_NOT_ALLOWED_NO_RX	RF_NOT_ALLOWED_NO_RX 0b - Assertion on RF_NOT_ALLOWED has no effect on RX 1b - Assertion on RF_NOT_ALLOWED can abort RX
4 RF_NOT_ALLOWED_NO_TX	RF_NOT_ALLOWED_NO_TX 0b - Assertion on RF_NOT_ALLOWED has no effect on TX 1b - Assertion on RF_NOT_ALLOWED can abort TX
3-0 RF_NOT_ALLOWED_EN	RF_NOT_ALLOWED PER-LINK-LAYER ENABLE The coexistence input RF_NOT_ALLOWED can be enabled to selectively abort TX or RX sequences, with individual enables for each supported protocol, subject also to the state of the RF_NOT_ALLOWED_NO_TX and RF_NOT_ALLOWED_NO_RX control bits, according to the following table: **xxx1**: RF_NOT_ALLOWED assertions are enabled to abort BLE TX and RX sequences **xxx0**: RF_NOT_ALLOWED assertions are not enabled to abort BLE TX and RX sequences **1xxx**: RF_NOT_ALLOWED assertions are enabled to abort GENERIC_FSK TX and RX sequences **0xxx**: RF_NOT_ALLOWED assertions are not enabled to abort GENERIC_FSK TX and RX sequences **NOTE:** RF_NOT_ALLOWED_EN[1] and RF_NOT_ALLOWED_EN[2] currently have no functionality

A.2.5.13 CRC/WHITENER CONFIG REGISTER (CRCW_CFG)

A.2.5.13.1 Offset

Register	Offset
CRCW_CFG	30h

A.2.5.13.2 Diagram



A.2.5.13.3 Fields

Field	Function
31-30 —	Reserved
29 CRC_EC_FAIL	CRC error correction fail This signal is cleared when *crc_init* is asserted. It is set after the completion of a CRC error correction calculation that does not find a valid data correction solution or if error correction was disabled and the CRC check found an error.
28 CRC_EC_DONE	CRC error correction done This signal is cleared when *crc_init* is asserted. It is set after the error correction logic completes processing the syndrome. It is immediately set after a packet is received if no error was detected or if error correction is disabled. Otherwise, it can take up to N system clocks after the packet is received to assert, with N = (number of bits used in the CRC calculation, including the CRC value).
27 —	Reserved
26-16 CRC_EC_OFFSET	CRC error correction offset This value provides the byte offset within the data packet to which the CRC error correction mask should be XOR-ed. This value is valid if the *crc_ec_done* signal is asserted and the *crc_ec_fail* signal is not asserted. The offset includes only the bytes used in the CRC calculation, including the CRC value.
15-5	Reserved

Table continues on the next page...

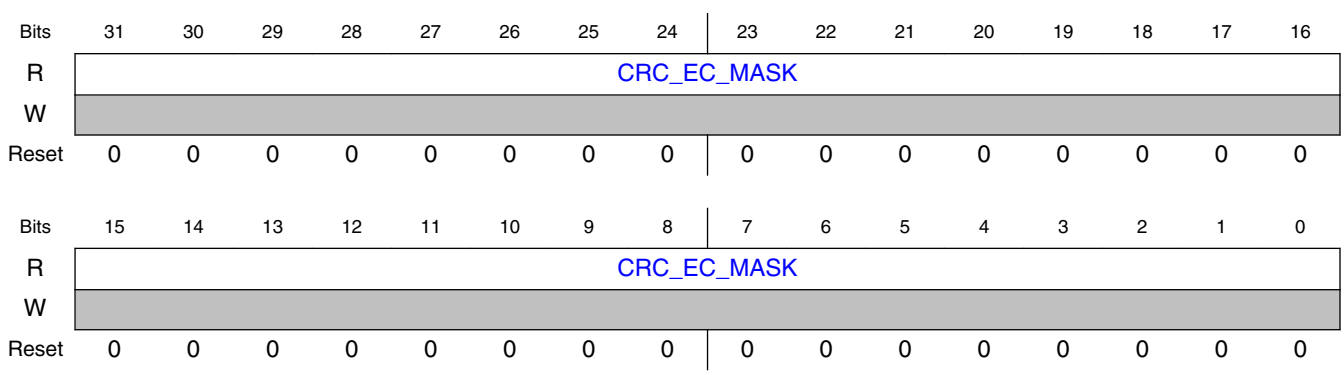
Field	Function
—	
4 CRC_RES_OUT_VLD	CRC result output valid CRC result output valid
3 CRC_EARLY_FAIL	CRC error correction fail This signal is cleared when *crc_init* is asserted. It is set after the completion of a CRC error correction calculation that does not find a valid data correction solution or if error correction was disabled and the CRC check found an error.
2 CRC_ZERO	CRC zero This signal is asserted at any time that the CRC shift register contains a value of zero
1 CRCW_EC_EN	CRC Error Correction Enable CRC Error Correction Enable
0 CRCW_EN	CRC calculation enable Input data bits loaded with this signal asserted are used in the CRC calculation.

A.2.5.14 CRC ERROR CORRECTION MASK (CRC_EC_MASK)

A.2.5.14.1 Offset

Register	Offset
CRC_EC_MASK	34h

A.2.5.14.2 Diagram



A.2.5.14.3 Fields

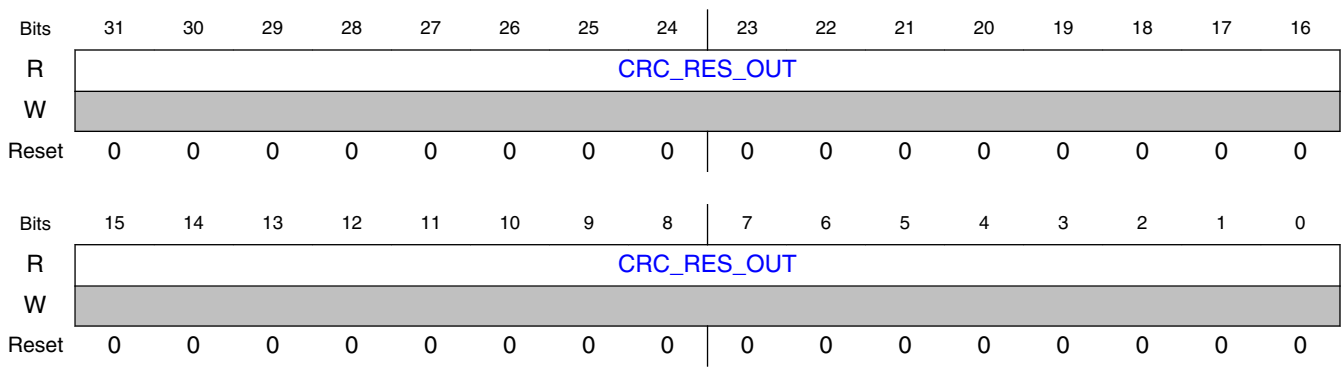
Field	Function
31-0 CRC_EC_MASK	CRC error correction mask This value provides a 32-bit XOR mask that must be applied to the input data packet to correct the burst errors detected by the error correction calculation. This value is valid if the *crc_ec_done* signal is asserted and the *crc_ec_fail* signal is not asserted.

A.2.5.15 CRC RESULT (CRC_RES_OUT)

A.2.5.15.1 Offset

Register	Offset
CRC_RES_OUT	38h

A.2.5.15.2 Diagram



A.2.5.15.3 Fields

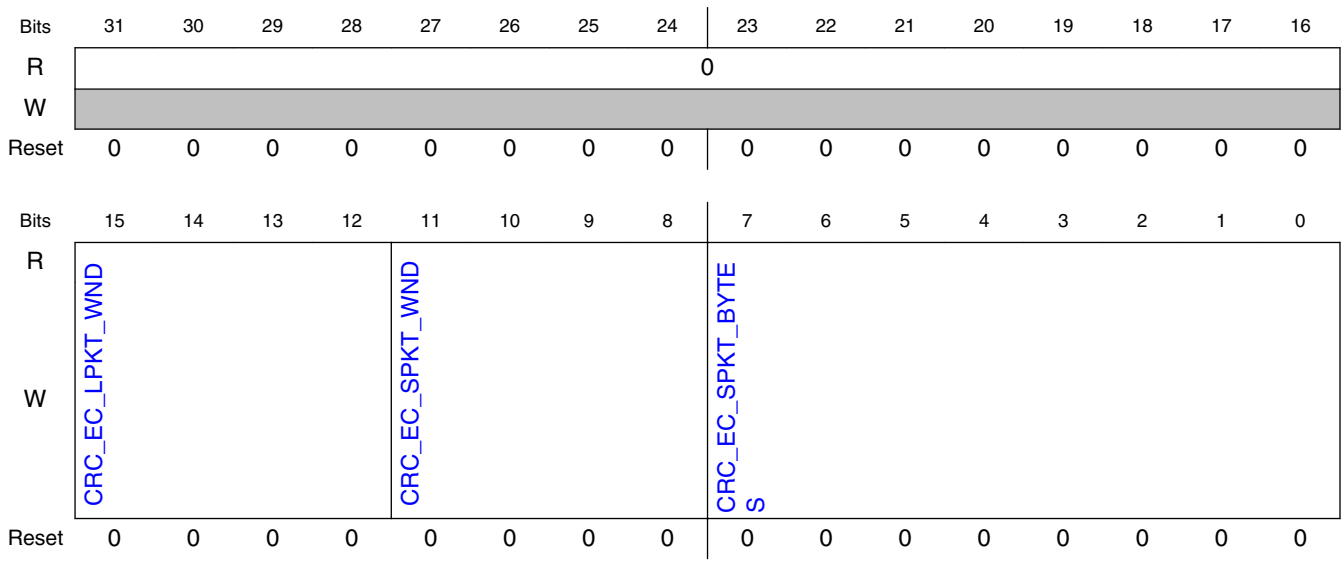
Field	Function
31-0 CRC_RES_OUT	CRC result output This bus provides the instantaneous value of the CRC shift register.

A.2.5.16 CRC/WHITENER CONFIG 2 REGISTER (CRCW_CFG2)

A.2.5.16.1 Offset

Register	Offset
CRCW_CFG2	3Ch

A.2.5.16.2 Diagram



A.2.5.16.3 Fields

Field	Function
31-16 —	Reserved
15-12 CRC_EC_LPKT_WND	Error correction long packet burst error window This value determines the largest burst error corrected when long packets are processed. A value of zero will disable long packet error correction. The max value is 15.
11-8 CRC_EC_SPKT_WND	Error correction short packet burst error window This value determines the largest burst error corrected when short packets are processed. A value of zero will disable short packet error correction. The max value is 15.
7-0 CRC_EC_SPKT_BYTES	Error Correction Short Packet Bytes Short packets are defined as those where the number of bytes used in the CRC calculation, including the CRC value, do not exceed this value. The max value is 255.

A.2.6 XCVR_PHY register descriptions

Note

To decipher number formats such as ?ufix4en2? or ?sfix8en3?:

General form: [u/s]fix[Nb]e[n][Nx]

Where,

[u/s]	denotes whether the number is signed or not
fix	denotes that the number is fixed point, 2's complement
Nb	indicates total number of bits, including a sign bit if signed
e	indicates that the number is exponentiated like num x 2^(exponent)
n	indicates the exponent is negative

Table continues on the next page...

Nx	indicates the value of the exponent

For example :

- ufix4en2(1101) = 1101 x 2^(-2) = 11.01 = 3.25 (decimal)
- sfix8en3(10011011) = -1100101 x 2^(-3) = -1100.101 = -12.625 (decimal)

A.2.6.1 XCVR_RX_PHY_ADDR Memory map

XCVR_PHY base address: 4005_C400h

Offset	Register	Width (In bits)	Access	Reset value
0h	PREAMBLE REFERENCE WAVEFORM 0 (PRE_REF0)	32	RW	FDDD_EB39h
4h	PREAMBLE REFERENCE WAVEFORM 1 (PRE_REF1)	32	RW	BEFB_FFFFh
8h	PREAMBLE REFERENCE WAVEFORM 2 (PRE_REF2)	32	RW	0000_CE75h
20h	PHY CONFIGURATION REGISTER 1 (CFG1)	32	RW	1070_CD16h
24h	PHY CONFIGURATION REGISTER 2 (CFG2)	32	RW	0100_0A48h
28h	PHY EARLY/LATE CONFIGURATION REGISTER (EL_CFG)	32	RW	0000_0000h
2Ch	PHY NETWORK ADDRESS FOR BSM (NTW_ADR_BSM)	32	RW	0000_0000h
30h	PHY STATUS REGISTER (STATUS)	32	RO	0000_0000h

A.2.6.2 PREAMBLE REFERENCE WAVEFORM 0 (PRE_REF0)

A.2.6.2.1 Offset

Register	Offset
PRE_REF0	0h

A.2.6.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FSK_PREAMBLE_REF0															
Reset	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FSK_PREAMBLE_REF0															
Reset	1	1	1	0	1	0	1	1	0	0	1	1	1	0	0	1

A.2.6.2.3 Fields

Field	Function
31-0 FSK_PREAMBLE_REF0	Base preamble reference waveform containing sixteen 5-bit phase values represented in 2's complement notation using 1 sign bit and 4 fractional bits. Thus, the range of representable values is [-1, 1-1/16] and they correspond to phases in the range [-π, π(1-1/16)]. Preamble reference waveform is an 80-bit vector; FSK_PREAMBLE_REF0 constitutes the lowest (least significant) component. The entire reference waveform is assembled by concatenating the following register values: Reference Waveform = {FSK_PREAMBLE_REF2[15:0], FSK_PREAMBLE_REF1[31:0], FSK_PREAMBLE_REF0[31:0] The preamble reference waveforms for various PHY configurations are as follows:

Generic FSK MODE	FSK_PREAMBLE-REF0	FSK_PREAMBLE_REF1	FSK_PREAMBLE_REF2
BLE (GFSK BT=0.5, h=0.5)	0x79CDEB39	0xCE77DEF7	0x0000CEB7
GFSK BT=0.5, h=0.32	0xBBDE739B	0xDEFBDEF7	0x0000E739
GFSK BT=0.5, h=0.7	0x37ACE2F7	0xADF3BDEF	0x0000BE33
GFSK BT=0.5, h=1.0	0xF38B5273	0x8CEF9CE6	0x00009D2D
GFSK BT=0.3, h=0.5	0x7BCDEB39	0xCEF7DEF7	0x0000CEB7
GFSK BT=0.7, h=0.5	0x79CDEB39	0xCE77DEF7	0x0000CEB7
MSK	0x79CDEB38	0xCE77DFF7	0x0000CEB7

A.2.6.3 PREAMBLE REFERENCE WAVEFORM 1 (PRE_REF1)

A.2.6.3.1 Offset

Register	Offset
PRE_REF1	4h

A.2.6.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FSK_PREAMBLE_REF1															
Reset	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FSK_PREAMBLE_REF1															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

A.2.6.3.3 Fields

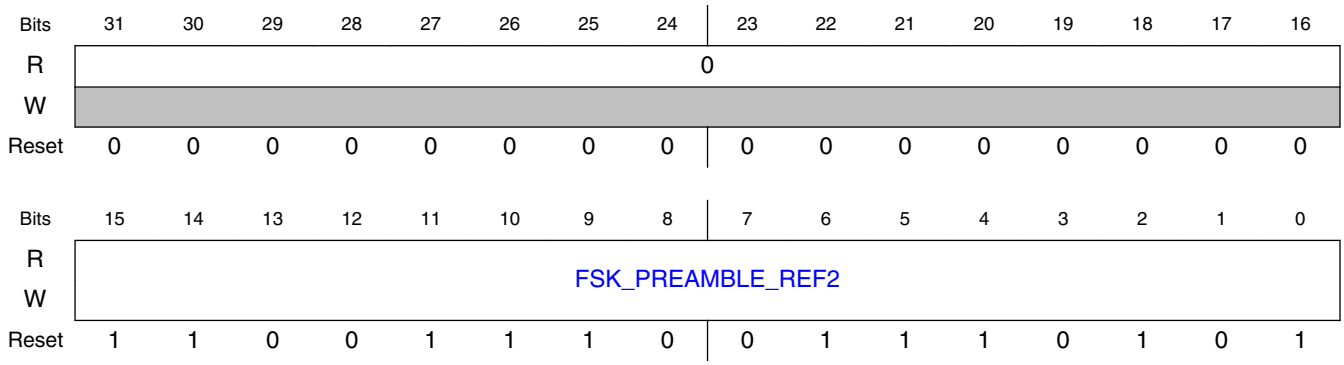
Field	Function
31-0 FSK_PREAMBLE_REF1	Refer to FSK_PREAMBLE_REF0.

A.2.6.4 PREAMBLE REFERENCE WAVEFORM 2 (PRE_REF2)

A.2.6.4.1 Offset

Register	Offset
PRE_REF2	8h

A.2.6.4.2 Diagram



A.2.6.4.3 Fields

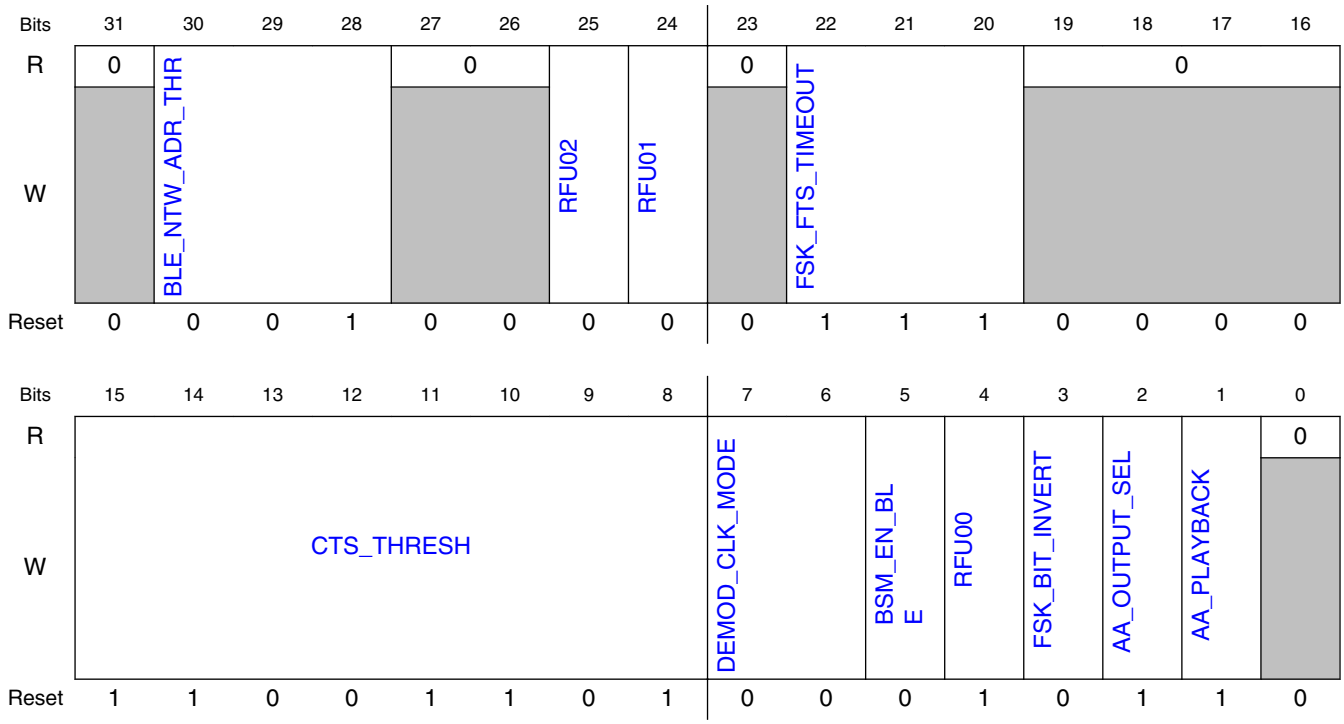
Field	Function
31-16 —	Reserved
15-0 FSK_PREAMBLE_REF2	Refer to FSK_PREAMBLE_REF0.

A.2.6.5 PHY CONFIGURATION REGISTER 1 (CFG1)

A.2.6.5.1 Offset

Register	Offset
CFG1	20h

A.2.6.5.2 Diagram



A.2.6.5.3 Fields

Field	Function
31 —	Reserved
30-28 BLE_NTW_ADR_THR	BLE Network Address Match Bit Error Threshold Number of Tolerated bit errors for Access Address correlation in BLE mode
27-26 —	Reserved
25 RFU02	Reserved for future use.
24 RFU01	Reserved for future use.
23 —	Reserved
22-20 FSK_FTS_TIMEOUT	FSK FTS Timeout Number of symbols FTS is allowed to proceed beyond the expected end of the longest AA in FSK 000b - 4 symbols 001b - 5 symbols 010b - 6 symbols

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>011b - 7 symbols
100b - 8 symbols
101b - 9 symbols
110b - 10 symbols
111b - 11 symbols</td></tr>
<tr><td>19-16
—</td><td>Reserved</td></tr>
<tr><td>15-8
CTS_THRESH</td><td>CTS Correlation Threshold
Coarse Timing Search (CTS) correlation threshold is an unsigned 8-bit fixed-point number that reprsents a range of CTS threshold representable values in [0, ..., 1-1/2^8] = {0000 0000, ..., 1111 1111}. CTS Threshold is a function of the PHY modulation scheme, data rate, and the receiver chain filtering characteristics and is chosen to maximize the preformance.
The table below shows the threshold values for an assortment of FSK schemes. They are determined from Simulink model simulations. For more details, please refer to the Multi-PHY chapter in the 2.4GHz Radio 2 ADD.
<table>
<tr><th>Modulation Scheme</th><th>Data Rate (kbps)</th><th>CTS_THRESH value</th><th>CTS_THRESH bit representation</th></tr>
<tr><td>BLE (GFSK BT=0.5, h=0.5)</td><td>1000</td><td>0.7500</td><td>1100 0000</td></tr>
<tr><td>ANT (GFSK BT=0.5, h=0.32)</td><td>1000</td><td>0.8008</td><td>1100 1101</td></tr>
<tr><td>GFSK BT=0.5, h=0.7</td><td>500</td><td>0.7500</td><td>1100 0000</td></tr>
<tr><td>GFSK BT=0.3, h=0.5</td><td>1000</td><td>0.8516</td><td>1101 1010</td></tr>
<tr><td>MSK</td><td>500</td><td>0.8125</td><td>1101 0000</td></tr>
<tr><td>GFSK BT=0.5, h=0.5</td><td>250</td><td>0.8750</td><td>1110 0000</td></tr>
<tr><td>GFSK BT=0.5, h=1.0</td><td>250</td><td>0.6875</td><td>1011 0000</td></tr>
<tr><td>GFSK BT=0.7, h=0.5</td><td>1000</td><td>0.8515</td><td>1101 1010</td></tr>
</table></td></tr>
<tr><td>7-6
DEMOD_CLK_MODE</td><td>Demodulator Clock Mode
00b - Normal
01b - Demodulate all samples
10b - Reserved
11b - Reserved</td></tr>
<tr><td>5
BSM_EN_BLE</td><td>BLE Bit Streaming Mode Enable bit
Enable the serialized, received BLE packet bitstream to appear on the BSM pins of the SoC. (See the XCVR BSM Block Guide)
0b - BSM for BLE disabled
1b - BSM for BLE enabled</td></tr>
<tr><td>4
RFU00</td><td>Reserved for future use.</td></tr>
<tr><td>3
FSK_BIT_INVERT</td><td>FSK Bit Invert
Inverts FSK mapping of symbols to bits when asserted.
<table>
<tr><th>FSK_BIT_INVERT</th><th>$E_s \geq 0$</th><th>$E_s < 0$</th></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
</table></td></tr>
</table>

Table continues on the next page...

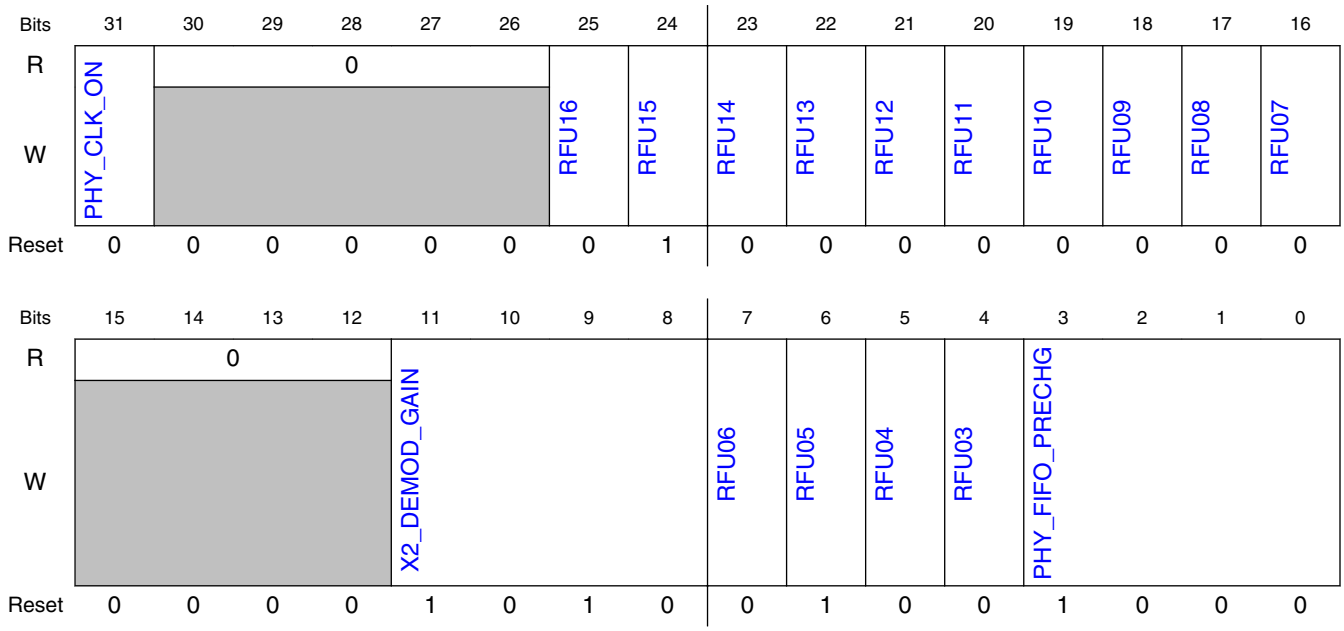
<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
<table>
<tr><th>FSK_BIT_INVERT</th><th>$E_s \geq 0$</th><th>$E_s < 0$</th></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
</table>
In the table above, E_s is the soft bit output from the symbol demodulator.
</td></tr>
<tr><td>2
AA_OUTPUT_SEL</td><td>Access Address Output Select
Selects whether the demodulated AA bit sequence or the matched AA pattern is output.
0b - demodulated
1b - matched</td></tr>
<tr><td>1
AA_PLAYBACK</td><td>Access Address Playback
Enable/disable output of Access Address bit sequence via data_out port depending on aa_output_sel according to the following table:
<table>
<tr><th>AA_PLAYBACK</th><th>AA_OUTPUT_SEL</th><th>ACTIONUNITS</th></tr>
<tr><td>0</td><td>X</td><td>Only PDU bits output via data_out port. No AA bits</td></tr>
<tr><td>1</td><td>0</td><td>Demodulated AA bits followed by PDU bits are output via data_out port.</td></tr>
<tr><td>1</td><td>1</td><td>Matched AA bits followed by PDU bits are output via data_out port.</td></tr>
</table>
Note: This bit must be set to 1 for Generic FSK, and must be set to 0 for BLE.</td></tr>
<tr><td>0
—</td><td>Reserved</td></tr>
</table>

A.2.6.6 PHY CONFIGURATION REGISTER 2 (CFG2)

A.2.6.6.1 Offset

Register	Offset
CFG2	24h

A.2.6.6.2 Diagram



A.2.6.6.3 Fields

Field	Function
31 PHY_CLK_ON	Force PHY Clock On (testmode) 0b - PHY clock is enabled by TSM output: rx_phy_en 1b - PHY clock is forced on at all times
30-26 —	Reserved
25 RFU16	Reserved for future use.
24 RFU15	Reserved for future use.
23 RFU14	Reserved for future use.
22 RFU13	Reserved for future use.
21 RFU12	Reserved for future use.
20 RFU11	Reserved for future use.
19 RFU10	Reserved for future use.
18	Reserved for future use.

Table continues on the next page...

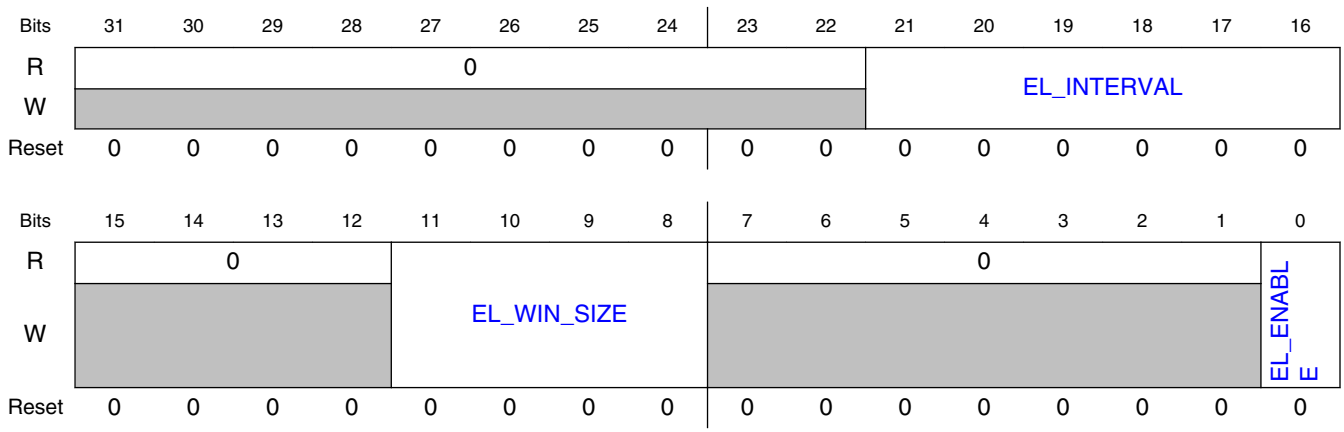
<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>RFU09</td><td></td></tr>
<tr><td>17
RFU08</td><td>Reserved for future use.</td></tr>
<tr><td>16
RFU07</td><td>Reserved for future use.</td></tr>
<tr><td>15-12
—</td><td>Reserved</td></tr>
<tr><td>11-8
X2_DEMOD_GAIN</td><td>X2_DEMOD_GAIN
Gain parameter used in the symbol demodulator. The unsigned fixed-point gain is 4 bits wide representing a range of representable gain values: [0, ..., 1-1/2^4] = {0000, ..., 1111}. X2_DEMOD_GAIN is a function of the PHY modulation, scheme, data rate, and the receiver chain filtering characteristics and is chosen to minimize the demodulation bit error rate.
The table below shows the recommended values for an assortment of FSK schemes. They are determined from Simulink model simulations. For more details, please refer to the Multi-PHY chapter in the 2.4GHz Radio Gen 2 ADD.
<table>
<tr><th>Modulation Scheme</th><th>Data Rate (kbps)</th><th>X2_DEMOD_GAIN value</th><th>X2_DEMOD_GAIN bit representation</th></tr>
<tr><td>BLE (GFSK BT=0.5, h=0.5)</td><td>1000</td><td>0.6250</td><td>1010</td></tr>
<tr><td>ANT (GFSK BT=0.5, h=0.32)</td><td>1000</td><td>0.1875</td><td>0011</td></tr>
<tr><td>GFSK BT=0.5, h=0.7</td><td>500</td><td>0.6250</td><td>1010</td></tr>
<tr><td>GFSK BT=0.3, h=0.5</td><td>1000</td><td>0.6250</td><td>1010</td></tr>
<tr><td>MSK</td><td>500</td><td>0.6250</td><td>1010</td></tr>
<tr><td>GFSK BT=0.5, h=0.5</td><td>250</td><td>0.2500</td><td>0100</td></tr>
<tr><td>GFSK BT=0.5, h=1.0</td><td>250</td><td>0.5000</td><td>1000</td></tr>
<tr><td>GFSK BT=0.7, h=0.5</td><td>1000</td><td>0.5000</td><td>1000</td></tr>
</table></td></tr>
<tr><td>7
RFU06</td><td>Reserved for future use.</td></tr>
<tr><td>6
RFU05</td><td>Reserved for future use.</td></tr>
<tr><td>5
RFU04</td><td>Reserved for future use.</td></tr>
<tr><td>4
RFU03</td><td>Reserved for future use.</td></tr>
<tr><td>3-0
PHY_FIFO_PRECHG</td><td>PHY FIFO Precharge Level
Indicates the precharge depth of the output FIFO.</td></tr>
</table>

A.2.6.7 PHY EARLY/LATE CONFIGURATION REGISTER (EL_CFG)

A.2.6.7.1 Offset

Register	Offset
EL_CFG	28h

A.2.6.7.2 Diagram



A.2.6.7.3 Fields

Field	Function
31-22 —	Reserved
21-16 EL_INTERVAL	EL_INTERVAL No. of FSK/IEEE 802.15.4 symbols between successive EL operation windows. Valid in both FSK and IEEE 802.15.4 modes. **Parameter** / **BLE** / **802.15.4** EL_WIN_SIZE[3:0] / 8 / 3 EL_INTERVAL[5:0] / 32 / 7
15-12 —	Reserved
11-8 EL_WIN_SIZE	EL_WIN_SIZE Number of successive FSK/IEEE 802.15.4 symbols over which one EL operation occurs. Valid in both FSK and IEEE 802.15.4 modes.
7-1	Reserved

Table continues on the next page...

Field	Function
—	
0 EL_ENABLE	EL_ENABLE Enable/disable EL mechanism during PDU/PSDU demodulation 0b - Disable Early/Late 1b - Enable Early/Late

A.2.6.8 PHY NETWORK ADDRESS FOR BSM (NTW_ADR_BSM)

A.2.6.8.1 Offset

Register	Offset
NTW_ADR_BSM	2Ch

A.2.6.8.2 Diagram

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

A.2.6.8.3 Fields

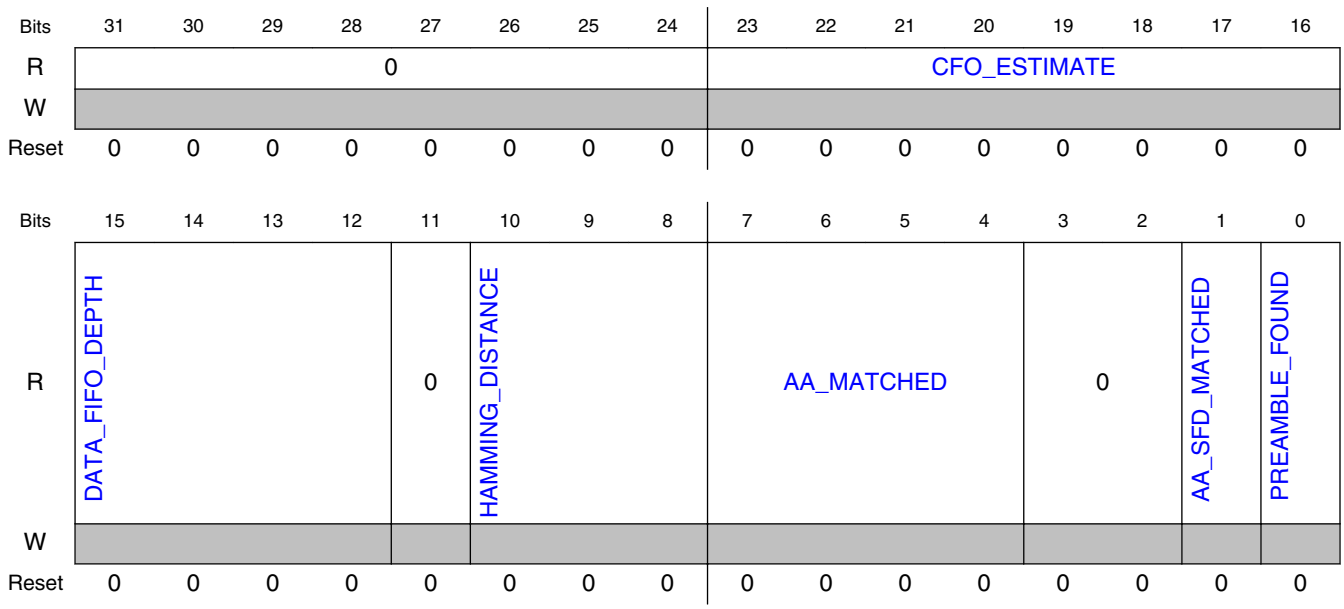
Field	Function
31-0 NTW_ADR_BSM	NTW_ADR_BSM PHY will search for this 32-bit Access Address when PHY_CFG1[BSM_EN_BLE]=1

A.2.6.9 PHY STATUS REGISTER (STATUS)

A.2.6.9.1 Offset

Register	Offset
STATUS	30h

A.2.6.9.2 Diagram



A.2.6.9.3 Fields

Field	Function
31-24 —	Reserved
23-16 CFO_ESTIMATE	Carrier Frequency Offset Estimate Most recent estimate for Carrier Frequency Offset. The multiplication factor is 7812: i.e., CFO_ESTIMATE x 7812 = actually frequency offset in Hz (ideally).
15-12 DATA_FIFO_DEPTH	DATA FIFO DEPTH Instantaneous depth of the PHY output FIFO. The difference between the FIFO write pointer and read pointer.
11 —	Reserved
10-8 HAMMING_DISTANCE	HAMMING DISTANCE Valid only in FSK mode. Indicates hamming distance between observed AA pattern and the candidate AA pattern that was found to be the best match.
7-4	Access Address Matched

Table continues on the next page...

Field	Function
AA_MATCHED	All bits reset in IDLE state. When any bit asserted, indicates which of the 4 Network Addresses has been matched. found. Valid only in FSK mode. 0000b - No Network Address has matched 0001b - Network Address 0 has matched 0010b - Network Address 1 has matched 0100b - Network Address 2 has matched 1000b - Network Address 3 has matched
3-2 —	Reserved
1 AA_SFD_MATCHED	Access Address or SFD Found Reset in IDLE state or when activate_search is de-asserted. Asserted when the AA/SFD is found.
0 PREAMBLE_FOUND	Preamble Found Reset in IDLE state or when *activate_search* is de-asserted. Asserted when the preamble is found and the coarse symbol timing is determined. If the subsequent AA/SFD search fails and the receiver resumes CTS, this signal will be reset.

A.2.7 XCVR_ANALOG register descriptions

A.2.7.1 XCVR_ANALOG_ADDR Memory map

XCVR_ANALOG base address: 4005_C500h

Offset	Register	Width (In bits)	Access	Reset value
0h	RF Analog Baseband LDO Control 1 (BB_LDO_1)	32	RW	0000_0000h
4h	RF Analog Baseband LDO Control 2 (BB_LDO_2)	32	RW	0000_0000h
8h	RF Analog ADC Control (RX_ADC)	32	RW	0000_0140h
Ch	RF Analog BBA Control (RX_BBA)	32	RW	0300_0003h
10h	RF Analog LNA Control (RX_LNA)	32	RW	0000_0000h
14h	RF Analog TZA Control (RX_TZA)	32	RW	0000_0003h
18h	RF Analog Aux PLL Control (RX_AUXPLL)	32	RW	0000_9002h
1Ch	RF Analog Synthesizer Control 1 (SY_CTRL_1)	32	RW	0000_0150h
20h	RF Analog Synthesizer Control 2 (SY_CTRL_2)	32	RW	0000_0014h
24h	RF Analog TX HPM DAC and PA Control (TX_DAC_PA)	32	RW	0002_0000h
28h	RF Analog Balun TX Mode Control (BALUN_TX)	32	RW	0072_4B6Dh
2Ch	RF Analog Balun RX Mode Control (BALUN_RX)	32	RW	0049_372Dh
30h	RF Analog DFT Observation Register 1 (DFT_OBSV_1)	32	RO	0000_0000h
34h	RF Analog DFT Observation Register 2 (DFT_OBSV_2)	32	RW	0000_0000h

Table continues on the next page...

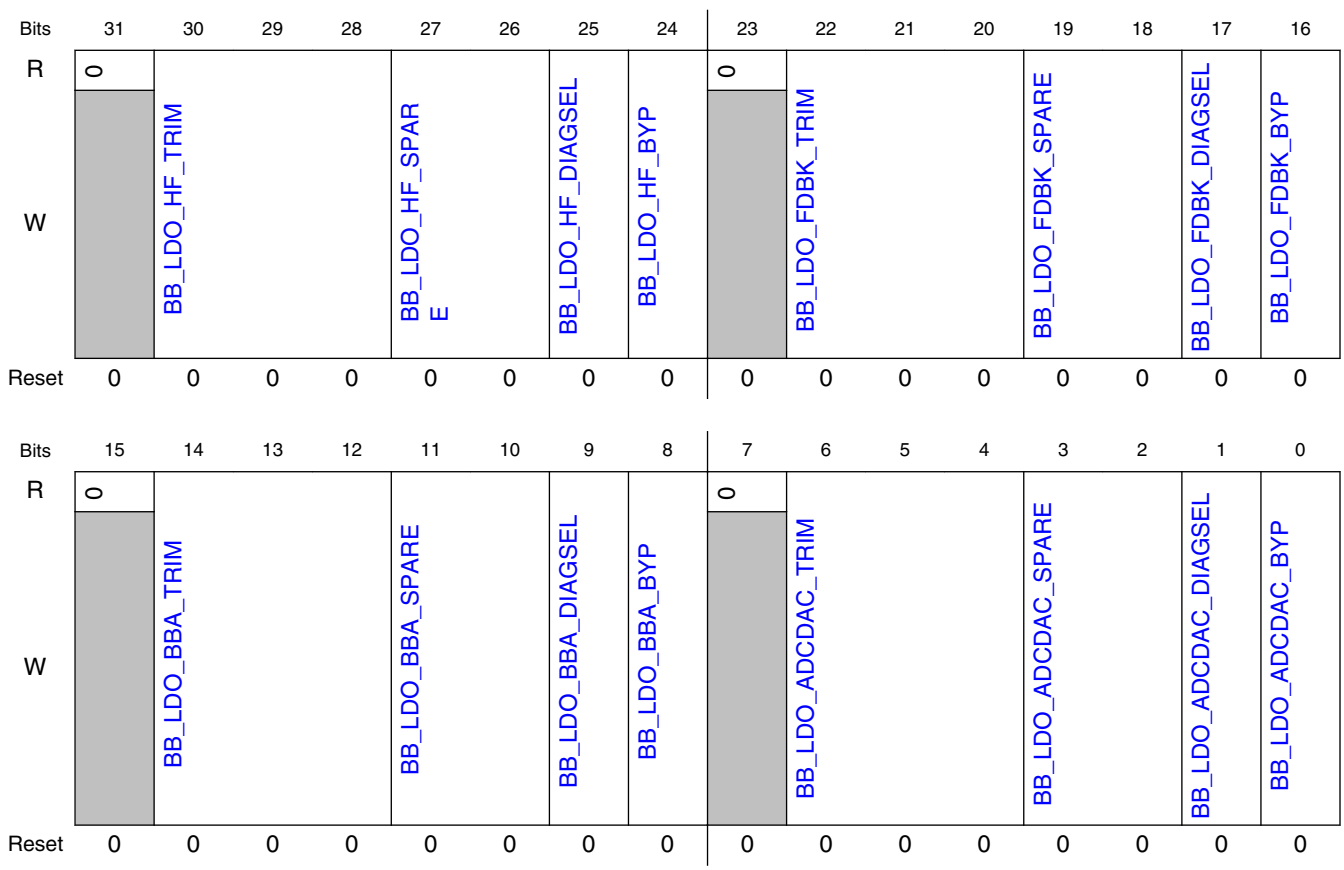
Offset	Register	Width (In bits)	Access	Reset value
38h	RF Analog DFT Observation Register 3 (DFT_OBSV_3)	32	RW	0000_FF00h

A.2.7.2 RF Analog Baseband LDO Control 1 (BB_LDO_1)

A.2.7.2.1 Offset

Register	Offset
BB_LDO_1	0h

A.2.7.2.2 Diagram



A.2.7.2.3 Fields

Field	Function
31 —	Reserved
30-28 BB_LDO_HF_TRIM	rmap_bb_ldo_hf_trim[2:0] Trim settings for LDO. 000b - 1.20 V (Default) 001b - 1.25 V 010b - 1.28 V 011b - 1.33 V 100b - 1.40 V 101b - 1.44 V 110b - 1.50 V 111b - 1.66 V
27-26 BB_LDO_HF_SPARE	rmap_bb_ldo_hf_spare[1:0] Spare bits. Not used so far.
25 BB_LDO_HF_DIAGSEL	rmap_bb_ldo_hf_diagsel Diagsel for LDO. 0b - Diag disable 1b - Diag enable
24 BB_LDO_HF_BYP	rmap_bb_ldo_hf_byp Bypass signal for LDO. Vline comes out directly. 0b - Bypass disabled. 1b - Bypass enabled
23 —	Reserved
22-20 BB_LDO_FDBK_TRIM	rmap_bb_ldo_fdbk_trim[2:0] Trim settings for LDO.(with 3.6/1.425V supply) 000b - 1.2/1.176 V (Default) 001b - 1.138/1.115 V 010b - 1.085/1.066 V 011b - 1.04/1.025 V 100b - 1.28/1.25 V 101b - 1.4/1.35 V 110b - 1.55/1.4 V 111b - 1.78/1.4 V
19-18 BB_LDO_FDBK_SPARE	rmap_bb_ldo_fdbk_spare[1:0] Spare bits
17 BB_LDO_FDBK_DIAGSEL	rmap_bb_ldo_fdbk_diagsel Diagsel for DIVN/FCAL LDO. 0b - Diag disable 1b - Diag enable
16	rmap_bb_ldo_fdbk_byp Bypass signal for DIVN/FCAL LDO. Vline comes out directly.

Table continues on the next page...

Field	Function
BB_LDO_FDBK_BYP	0b - Bypass disabled. 1b - Bypass enabled
15 —	Reserved
14-12 BB_LDO_BBA_TRIM	rmap_bb_ldo_bba_trim[2:0] Trim settings for LDO. 000b - 1.20 V (Default) 001b - 1.25 V 010b - 1.28 V 011b - 1.33 V 100b - 1.40 V 101b - 1.44 V 110b - 1.50 V 111b - 1.66 V
11-10 BB_LDO_BBA_SPARE	rmap_bb_ldo_bba_spare[1:0] Spare bits. Not used so far.
9 BB_LDO_BBA_DIAGSEL	rmap_bb_ldo_bba_diagsel Diagsel for LDO. 0b - Diag disable 1b - Diag enable
8 BB_LDO_BBA_BYP	rmap_bb_ldo_bba_byp Bypass signal for LDO. Vline comes out directly. 0b - Bypass disabled. 1b - Bypass enabled
7 —	Reserved
6-4 BB_LDO_ADCDAC_TRIM	rmap_bb_ldo_adcdac_trim[2:0] Trim settings for LDO. 000b - 1.20 V (Default) 001b - 1.25 V 010b - 1.28 V 011b - 1.33 V 100b - 1.40 V 101b - 1.44 V 110b - 1.50 V 111b - 1.66 V
3-2 BB_LDO_ADCDAC_SPARE	rmap_bb_ldo_adcdac_spare[1:0] Spare bits. Not used so far.
1 BB_LDO_ADCDAC_DIAGSEL	rmap_bb_ldo_adcdac_diagsel Diagsel for LDO. 0b - Diag disable 1b - Diag enable
0	rmap_bb_ldo_adcdac_byp Bypass signal for LDO. Vline comes out directly.

Field	Function
BB_LDO_ADCD AC_BYP	0b - Bypass disabled. 1b - Bypass enabled

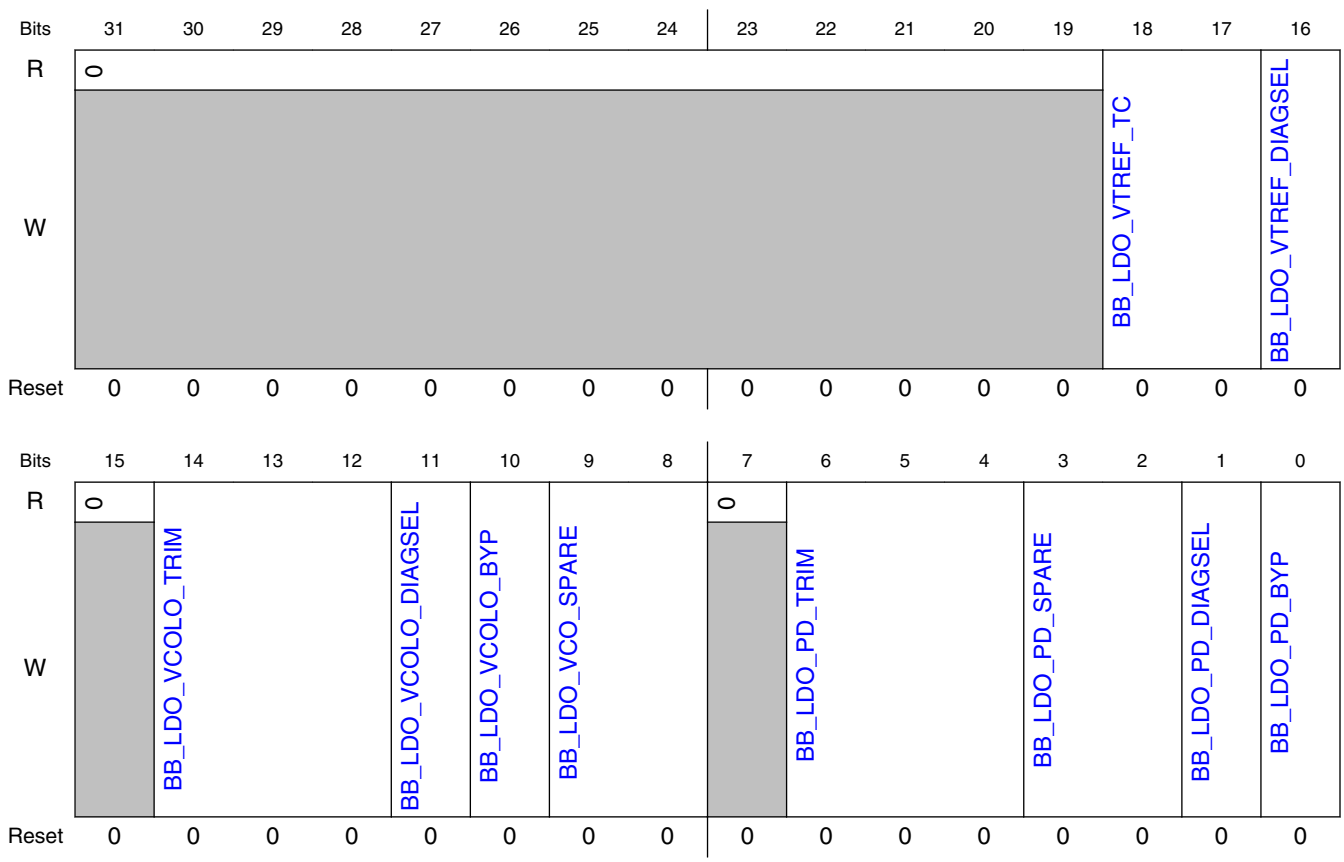
A.2.7.3 RF Analog Baseband LDO Control 2 (BB_LDO_2)

A.2.7.3.1 Offset

Register	Offset
BB_LDO_2	4h

A.2.7.3.2 Function

A.2.7.3.3 Diagram



A.2.7.3.4 Fields

Field	Function
31-19 —	Reserved
18-17 BB_LDO_VTREF_TC	rmap_bb_ldo_vtref_tc[1:0] Changes Temperature coefficient of reference(vldo_vcolo,vldo_fdbk) 00b - 1.117/1.176 V 01b - 1.134/1.188 V 10b - 1.10/1.162 V 11b - 1.09/1.152 V
16 BB_LDO_VTREF_DIAGSEL	rmap_bb_ldo_vtref_diagsel Diagsel for Reference. 0b - Diag disable 1b - Diag enable
15 —	Reserved
14-12 BB_LDO_VCOLO_TRIM	rmap_bb_ldo_vcolo_trim[2:0] Trim settings for LDO.(with 3.6/1.425V supply) 000b - 1.138/1.117 V (Default) 001b - 1.076/1.058 V 010b - 1.027/1.012 V 011b - 0.98/0.97 V 100b - 1.22/1.19 V 101b - 1.33/1.3 V 110b - 1.5/1.4 V 111b - 1.82/1.4 V
11 BB_LDO_VCOLO_DIAGSEL	rmap_bb_ldo_vcolo_diagsel Diagsel for VCO/LO LDO. 0b - Diag disable 1b - Diag enable
10 BB_LDO_VCOLO_BYP	rmap_bb_ldo_vcolo_byp Bypass signal for VCO/LO LDO. Vline comes out directly. 0b - Bypass disabled. 1b - Bypass enabled
9-8 BB_LDO_VCO_SPARE	rmap_bb_ldo_vco_spare[1:0] Spare bits
7 —	Reserved
6-4 BB_LDO_PD_TRIM	rmap_bb_ldo_pd_trim[2:0] Trim settings for LDO. 000b - 1.20 V (Default) 001b - 1.25 V 010b - 1.28 V 011b - 1.33 V

Table continues on the next page...

Field	Function
	100b - 1.40 V 101b - 1.44 V 110b - 1.50 V 111b - 1.66 V
3-2 BB_LDO_PD_SPARE	rmap_bb_ldo_pd_spare[1:0] Spare bits. Not used so far.
1 BB_LDO_PD_DIAGSEL	rmap_bb_ldo_pd_diagsel Diagsel for Reference. 0b - Diag disable 1b - Diag enable
0 BB_LDO_PD_BYP	rmap_bb_ldo_pd_byp Bypass signal for LDO. Vline comes out directly. 0b - Bypass disabled. 1b - Bypass enabled

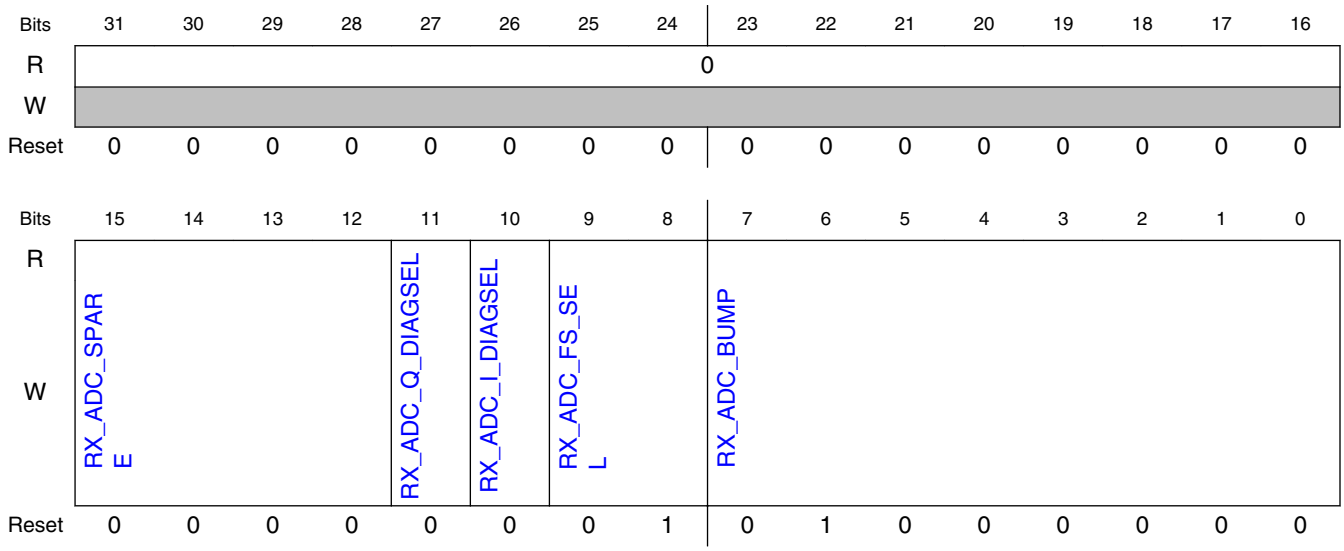
A.2.7.4 RF Analog ADC Control (RX_ADC)

A.2.7.4.1 Offset

Register	Offset
RX_ADC	8h

A.2.7.4.2 Function

A.2.7.4.3 Diagram



A.2.7.4.4 Fields

Field	Function
31-16 —	Reserved
15-12 RX_ADC_SPARE	rmap_rx_adc_spare[3:0] Not used
11 RX_ADC_Q_DIAGSEL	rmap_rx_adc_q_diagsel Enable diagnostics output for ADC Q
10 RX_ADC_I_DIAGSEL	rmap_rx_adc_i_diagsel Enable diagnostics output for ADC I
9-8 RX_ADC_FS_SEL	rmap_rx_adc_fs_sel[1:0] Clock frequency selector 00b - 52MHz (2x26MHz) 01b - 64MHz (2x32MHz) 10b - +13% of 64MHz 11b - - 11% of 64MHz
7-0 RX_ADC_BUMP	rmap_rx_adc_bump[7:0] Bump bits for both ADCs : bits [7:6] bump all opamps bias current 00 : 0.75x 01 : 1.00x

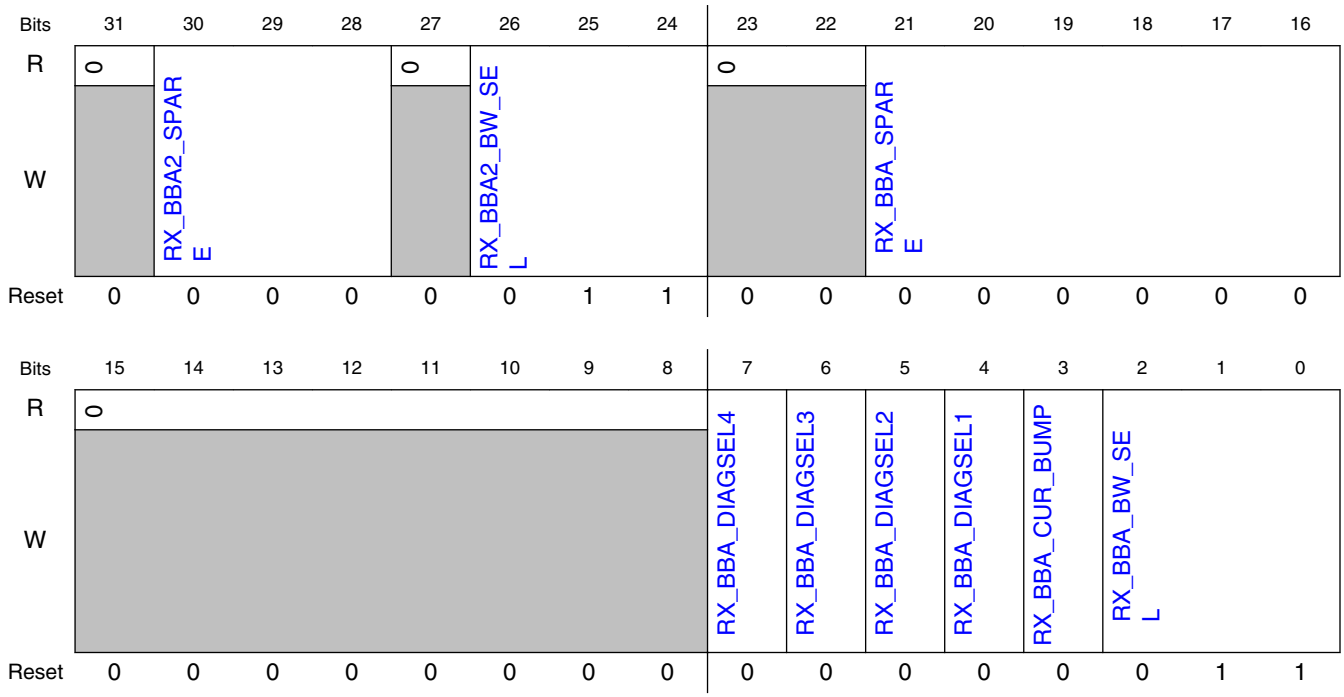
Field	Function
	10 : 1.25x 11 : 1.50x bit [5] bump dac opamp bias current. 0 : 1.0x 1 : 1.5x bit [4] bump dac vrefn 0 : 0.0V 1 : 0.1V bit [3] bump opamp CM output voltage 0 : vcmo = 0.63V 1 : vcmo = 0.57V bits [2:1] Reserved bit [0] bump Rg resistor 0 : resonator notch at nominal freq. 1 : resonator notch at 10% higher freq.

A.2.7.5 RF Analog BBA Control (RX_BBA)

A.2.7.5.1 Offset

Register	Offset
RX_BBA	Ch

A.2.7.5.2 Diagram



A.2.7.5.3 Fields

Field	Function
31 —	Reserved
30-28 RX_BBA2_SPARE	rmap_rx_bba2_spare[2:0] Spare bits reserved for future use.
27 —	Reserved
26-24 RX_BBA2_BW_SEL	rmap_bba2_bw_sel[2:0] Bandwidth control bit for baseband filter chain 000b - 1000K 001b - 900K 010b - 800K 011b - 700K Default 100b - 600K 101b - 500K
23-22 —	Reserved
21-16 RX_BBA_SPARE	rmap_rx_bba_spare[5:0] Spare bits reserved for future use. bit[5:2] are unused. bit[1:0] are mapped as follows for baseband common mode programming:

Table continues on the next page...

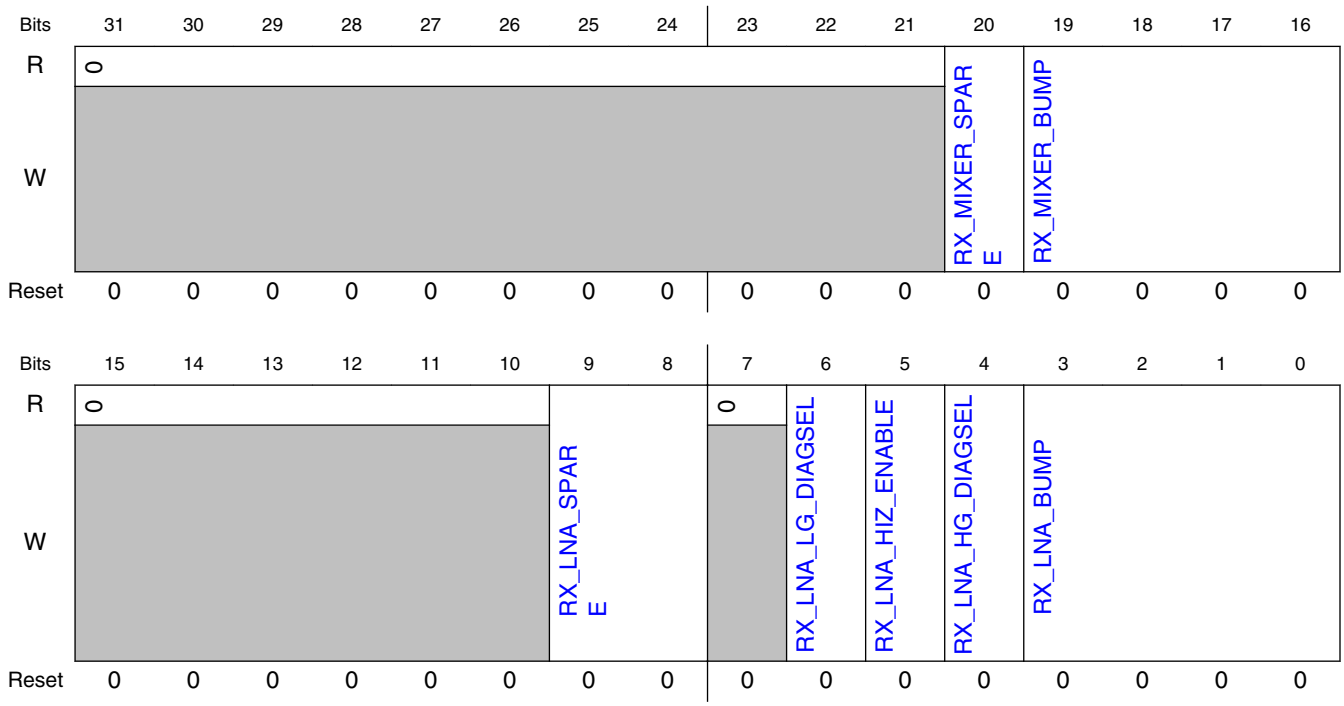
Field	Function
	000000b - 600mV (Default) 000001b - 675mV 000010b - 450mV 000011b - 525mV
15-8 —	Reserved
7 RX_BBA_DIAGSEL4	rmap_rx_bba_diagsel4 Diagsel for BBA 0b - Diag disable 1b - Diag enable
6 RX_BBA_DIAGSEL3	rmap_rx_bba_diagsel3 Diagsel for BBA 0b - Diag disable 1b - Diag enable
5 RX_BBA_DIAGSEL2	rmap_rx_bba_diagsel2 Diagsel for BBA 0b - Diag disable 1b - Diag enable
4 RX_BBA_DIAGSEL1	rmap_rx_bba_diagsel1 Diagsel for BBA 0b - Diag disable 1b - Diag enable
3 RX_BBA_CUR_BUMP	rmap_rx_bba_cur_bump BBA amplifier bias current setting. Default= 0. Setting to 1 increases amp current by 1.5X
2-0 RX_BBA_BW_SEL	rmap_rx_bba_bw_sel[2:0] Bandwidth control bit for baseband filter chain 000b - 1000K 001b - 900K 010b - 800K 011b - 700K Default 100b - 600K 101b - 500K

A.2.7.6 RF Analog LNA Control (RX_LNA)

A.2.7.6.1 Offset

Register	Offset
RX_LNA	10h

A.2.7.6.2 Diagram



A.2.7.6.3 Fields

Field	Function
31-21 —	Reserved
20 RX_MIXER_SPARE	rmap_rx_mixer_spare Spare bit
19-16 RX_MIXER_BUMP	rmap_rx_mixer_bump[3:0] [1:0] used as Bump bits for Mixer gate bias: [3:2] bits are unused. 0000b - 825mV (Default) 0001b - 750mV 0010b - 900mV 0011b - 975mV
15-10 —	Reserved
9-8 RX_LNA_SPARE	rmap_rx_lna_spare[1:0] Spare [1] is not used. Spare [0] is used to Bump the PTAT current generating resistor in LNA. Default is 0, Setting to a 1 increases the resistor by 5.6%

Table continues on the next page...

Field	Function
7 —	Reserved
6 RX_LNA_LG_DIAGSEL	rmap_rx_lna_lg_diagsel diagsel to check voltage bias in HG pathway circuits. (LNA bias/cmfb block and PTAT current gen).
5 RX_LNA_HIZ_ENABLE	rmap_rx_lna_hiZ_enable This bit doesn't exist in LNA top. When LNA is powered down, outputs are highZ.
4 RX_LNA_HG_DIAGSEL	rmap_rx_lna_hg_diagsel diagsel to check voltage bias in HG pathway circuits. (LNA bias/cmfb block and PTAT current gen).
3-0 RX_LNA_BUMP	rmap_rx_lna_bump[3:0] [1:0] used as Bump bits for LNA current: [3:2] used to Bump LNA output CM: 0000b - Default 0001b - -25% 0010b - +50% 0011b - +25% 0100b - CM 480mV 1000b - CM 600mV 1100b - CM 660mV

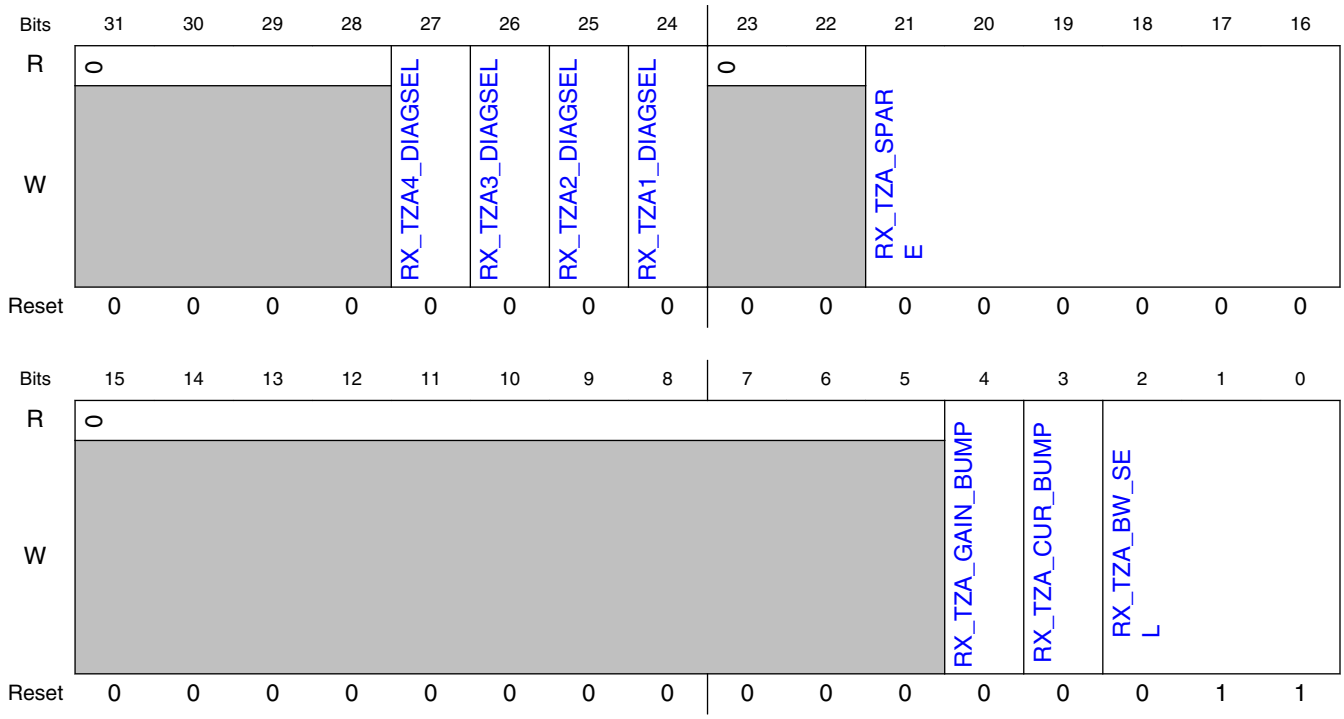
A.2.7.7 RF Analog TZA Control (RX_TZA)

A.2.7.7.1 Offset

Register	Offset
RX_TZA	14h

A.2.7.7.2 Function

A.2.7.7.3 Diagram



A.2.7.7.4 Fields

Field	Function
31-28 —	Reserved
27 RX_TZA4_DIAGSEL	rmap_rx_tza4_diagsel Diagsel for TZA 0b - Diag disable 1b - Diag enable
26 RX_TZA3_DIAGSEL	rmap_rx_tza3_diagsel Diagsel for TZA 0b - Diag disable 1b - Diag enable
25 RX_TZA2_DIAGSEL	rmap_rx_tza2_diagsel Diagsel for TZA 0b - Diag disable 1b - Diag enable
24 RX_TZA1_DIAGSEL	rmap_rx_tza1_diagsel Diagsel for TZA 0b - Diag disable 1b - Diag enable

Table continues on the next page...

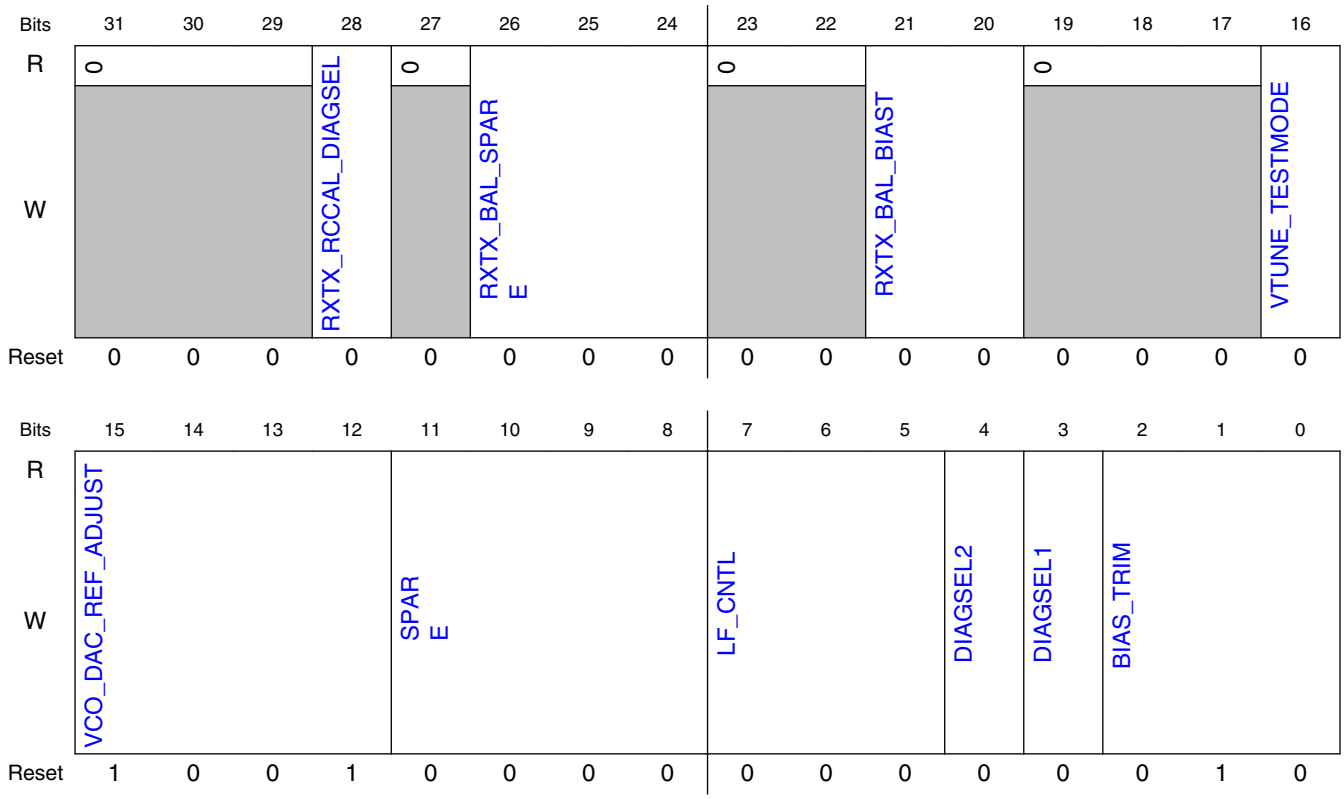
Field	Function
23-22 —	Reserved
21-16 RX_TZA_SPARE	rmap_rx_tza_spare[5:0] bit[5:3] are unused.bit[2] when clear uses a limited range for the TZA DCOC DAC; when set to 1 reverts to full range for TZA DCOC DAC. bit[1:0] are mapped as follows for baseband common mode programming: 000000b - 600mV (Default) 000001b - 675mV 000010b - 450mV 000011b - 525mV
15-5 —	Reserved
4 RX_TZA_GAIN_BUMP	rmap_rx_tza_gain_bump TZA Trans-impedance gain setting. Default=0. Setting it to 1 bumps DOWN the Trans-impedance of the TZA by 25%
3 RX_TZA_CUR_BUMP	rmap_rx_tza_cur_bump TZA amplifier bias current setting. Default= 0. Setting to 1 increases amp current by 1.5X
2-0 RX_TZA_BW_SEL	rmap_rx_tza_bw_sel[2:0] Bandwidth control bit for baseband filter chain 000b - 1000K 001b - 900K 010b - 800K 011b - 700K Default 100b - 600K 101b - 500K

A.2.7.8 RF Analog Aux PLL Control (RX_AUXPLL)

A.2.7.8.1 Offset

Register	Offset
RX_AUXPLL	18h

A.2.7.8.2 Diagram



A.2.7.8.3 Fields

Field	Function
31-29 —	Reserved
28 RXTX_RCCAL_DIAGSEL	rmap_rxtx_rccal_diagsel Enables diagnostics output
27 —	Reserved
26-24 RXTX_BAL_SPARE	rmap_rxtx_bal_spare[2:0] Spare bits. Not used so far.
23-22 —	Reserved
21-20 RXTX_BAL_BIAST	rmap_rxtx_bal_biast[1:0] Internal bias bump for the cap array. 00b - 0.6 01b - 0.4

Table continues on the next page...

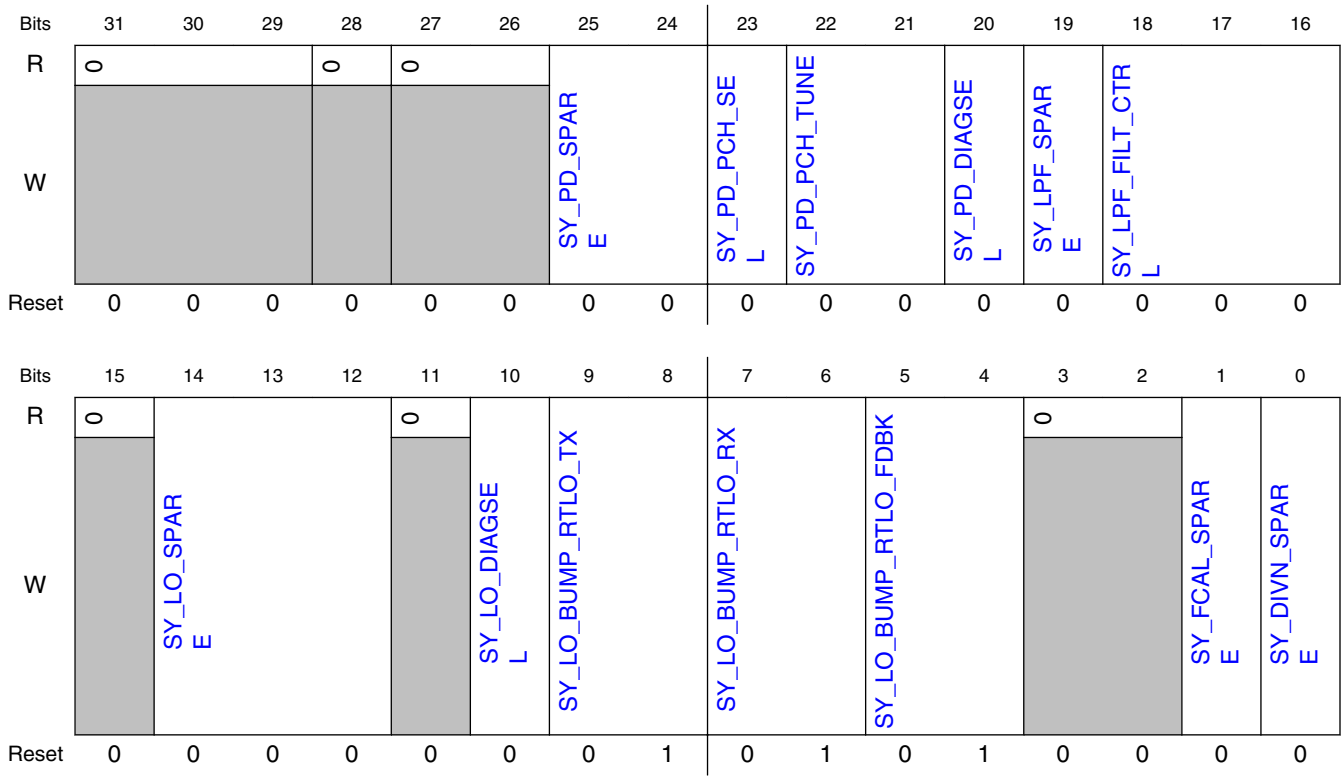
Field	Function
	10b - 0.9 11b - 1.2
19-17 —	Reserved
16 VTUNE_TESTMODE	rmap_rxtx_auxpll_vtune_testmode Testmode to force the vtune externally
15-12 VCO_DAC_REF_ADJUST	rmap_rxtx_auxpll_vco_dac_ref_adjust[3:0] Bits to tune the DAC current based on reference frequency of 26MHz or 32 MHz 4 = 26MHz RF_OSC recommended value. 7 = 32MHz RF_OSC recommended value. 9 = default value at reset, not recommended for use.
11-8 SPARE	rmap_rxtx_auxpll_spare[3:0] Spare bits - Spare [3] and Spare[2] are used as pins with following functions: Spare [3] - auxpll_clk_out_sel 0 selects the output of auxpll to analog(ADC) and digital 1 selects DSM clock from divN as the output to analog(ADC) and digital. Spare [2] 0 auxpll output to XO is disabled 1 The output of auxpll is sent to the XO to be sent out to xo output pad. Spare [0] is used to Bump the PTAT current generating resistor in AUXPLL. Default is 0, Setting to a 1 increases the resistor by 5.6%
7-5 LF_CNTL	rmap_rxtx_auxpll_lf_cntl[2:0] Bits to configure the low pass filter poles
4 DIAGSEL2	rmap_rxtx_auxpll_diagsel2 Diagnostic select 2
3 DIAGSEL1	rmap_rxtx_auxpll_diagsel1 Diagnostic select 1
2-0 BIAS_TRIM	rmap_rxtx_auxpll_bias_trim[2:0] Change the ratio in which the BG and PTAT currents are added in IREF block which supplies current to RO VCO

A.2.7.9 RF Analog Synthesizer Control 1 (SY_CTRL_1)

A.2.7.9.1 Offset

Register	Offset
SY_CTRL_1	1Ch

A.2.7.9.2 Diagram



A.2.7.9.3 Fields

Field	Function
31-29 —	Reserved
28 —	Reserved
27-26 —	Reserved
25-24 SY_PD_SPARE	rmap_sy_pd_spare[1:0] Spare[0] is used as pd_out pull down mode - 00b - Default ; 01b - PD output is pulled down.
23 SY_PD_PCH_SEL	rmap_sy_pd_pch_sel pch_sel is used to select the inverter based precharge or resistor divider based precharge 0b - inverter based precharge 1b - resistor divider based precharge

Table continues on the next page...

Field	Function
22-21 SY_PD_PCH_TUNE	rmap_sy_pd_pch_tune[1:0] These bits adjust the precharge voltage
20 SY_PD_DIAGSEL	rmap_sy_pd_diagsel Enables the diagnostic select
19 SY_LPF_SPARE	rmap_sy_lpf_spare Spare bits
18-16 SY_LPF_FILT_CTRL	rmap_sy_lpf_filt_ctrl[2:0] Configures the pole location and the order of the filer in LPF
15 —	Reserved
14-12 SY_LO_SPARE	rmap_sy_lo_spare[2:0] Spare bits
11 —	Reserved
10 SY_LO_DIAGSEL	rmap_sy_lo_diagsel Diagsel for LO. 0b - Diag disable 1b - Diag enable
9-8 SY_LO_BUMP_RTLO_TX	rmap_sy_lo_bump_rtlo_tx[1:0] Change resistance from supply to RX LO to change DC drop(vddl_div/vddl_buf) 00b - 1.071/1.065 V 01b - 1.092/1.090 V 10b - 1.099/1.098 V 11b - 1.10/1.1 V
7-6 SY_LO_BUMP_RTLO_RX	rmap_sy_lo_bump_rtlo_rx[1:0] Change resistance from supply to RX LO to change DC drop(vddl_div/vddl_buf) 00b - 1.051/1.037 V 01b - 1.082/1.075 V 10b - 1.092/1.088 V 11b - 1.098/1.094 V
5-4 SY_LO_BUMP_RTLO_FDBK	rmap_sy_lo_bump_rtlo_fdbk[1:0] Change resistance from supply to RX LO to change DC drop(vddl) 00b - 1.045 V 01b - 1.084 V 10b - 1.097 V 11b - 1.10 V
3-2 —	Reserved
1	rmap_sy_fcal_spare Spare bits

Table continues on the next page...

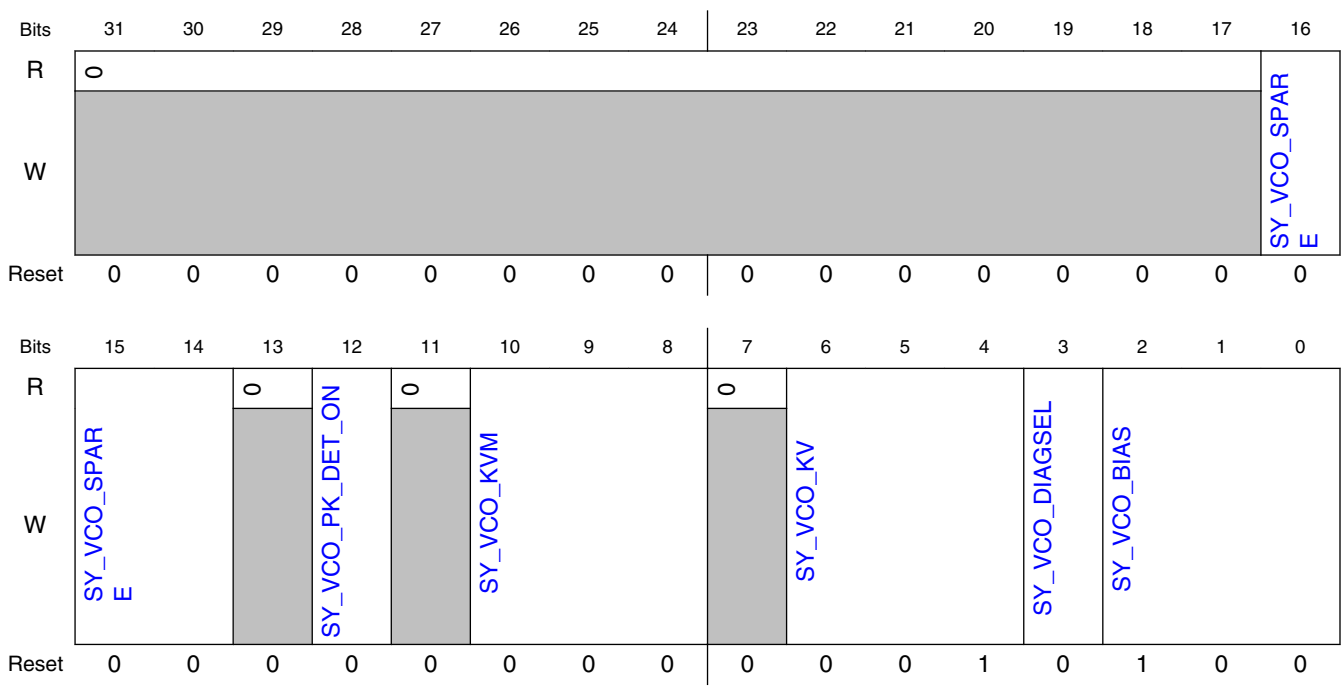
Field	Function
SY_FCAL_SPARE	
0 SY_DIVN_SPARE	rmap_sy_divn_spare Spare bits

A.2.7.10 RF Analog Synthesizer Control 2 (SY_CTRL_2)

A.2.7.10.1 Offset

Register	Offset
SY_CTRL_2	20h

A.2.7.10.2 Diagram



A.2.7.10.3 Fields

Field	Function
31-17	Reserved

Table continues on the next page...

Field	Function
—	
16-14 SY_VCO_SPARE	rmap_sy_vco_spare[2:0] Spare bits
13 —	Reserved
12 SY_VCO_PK_DET_ON	rmap_sy_vco_pk_det_on Enable peak detect for VCO. 0b - Disable 1b - Enable
11 —	Reserved
10-8 SY_VCO_KVM	rmap_sy_vco_kvm[2:0] Changes modulation port Kv. 000b - 10MHz/V 001b - 20MHz/V 010b - 30MHz/V 011b - 40MHz/V 100b - 40MHz/V 101b - 40MHz/V 110b - 40MHz/V 111b - 40MHz/V
7 —	Reserved
6-4 SY_VCO_KV	rmap_sy_vco_kv[2:0] Changes control port Kv. 000b - 50MHz/V 001b - 60MHz/V 010b - 70MHz/V 011b - 80MHz/V 100b - 80MHz/V 101b - 80MHz/V 110b - 80MHz/V 111b - 80MHz/V
3 SY_VCO_DIAGSEL	rmap_sy_vco_diagsel Diagsel for VCO. 0b - Diag disable 1b - Diag enable
2-0 SY_VCO_BIAS	rmap_sy_vco_bias[2:0] Change resistance from supply to VCO to change the DC drop (ldo supply = 1.124V and dc drop of 10mV) 000b - 0.97V 001b - 1.033V 010b - 1.06V 011b - 1.07V 100b - 1.08V 101b - 1.085V

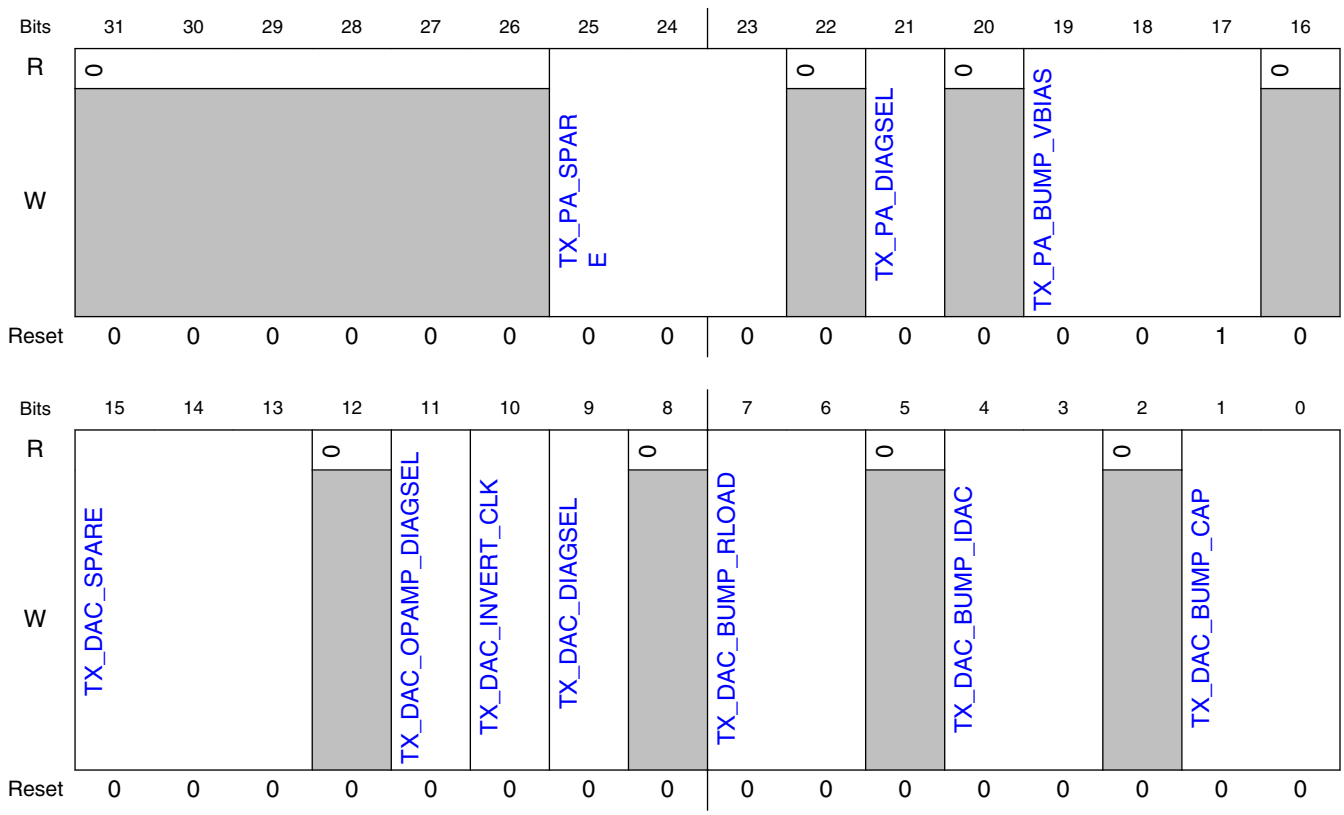
Field	Function
	110b - 1.090V 111b - 1.095V

A.2.7.11 RF Analog TX HPM DAC and PA Control (TX_DAC_PA)

A.2.7.11.1 Offset

Register	Offset
TX_DAC_PA	24h

A.2.7.11.2 Diagram



A.2.7.11.3 Fields

Field	Function
31-26	Reserved

Table continues on the next page...

Field	Function
—	
25-23 TX_PA_SPARE	rmap_tx_pa_spare[2:0] Spare bits. Not used so far. Spare bits [2:1] are not used. Spare [0] is used to Bump the PTAT current generating resistor in TX PA. Default is 0, Setting to a 1 increases the resistor by 5.6%
22 —	Reserved
21 TX_PA_DIAGSEL	rmap_tx_pa_diagsel Block level diag select bit
20 —	Reserved
19-17 TX_PA_BUMP_VBIAS	rmap_tx_pa_bump_vbias[2:0] Bump bits for the PA bias voltage for the gm/Nmos. 000b - 0.557 001b - 0.651 010b - 0.741 011b - 0.822 100b - 0.590 101b - 0.683 110b - 0.771 111b - 0.850
16 —	Reserved
15-13 TX_DAC_SPARE	rmap_tx_dac_spare[2:0] Spare bits for DAC/OPMAP
12 —	Reserved
11 TX_DAC_OPAMP_DIAGSEL	rmap_tx_dac_opamp_diagsel Enables Diagnostics for OPAMP 0b - Disable Diag 1b - Enable Diag
10 TX_DAC_INVERT_CLK	rmap_tx_dac_invert_clk
9 TX_DAC_DIAGSEL	rmap_tx_dac_diagsel Enables Diagnostics for DAC 0b - Disable Diag 1b - Enable Diag
8 —	Reserved
7-6	rmap_tx_dac_bump_rload[1:0]

Table continues on the next page...

Field	Function
TX_DAC_BUMP_RLOAD	change DAC resistor load. 00b - 3.12 kohms(default) 01b - 2.34 kohms 10b - 3.9 kohms 11b - 4.6 kohms
5 —	Reserved
4-3 TX_DAC_BUMP_IDAC	rmap_tx_dac_bump_idac[1:0] change DAC LSB current.(4*lsb current quoted here) 00b - 250nA(default) 01b - 207nA 10b - 312nA 11b - 415nA
2 —	Reserved
1-0 TX_DAC_BUMP_CAP	rmap_tx_dac_bump_cap[1:0] change filter cap at opamp to filter DAC images further. 00b - 1pF(default) 01b - 1.5pF 10b - 1.5pF 11b - 2pF

A.2.7.12 RF Analog Balun TX Mode Control (BALUN_TX)

A.2.7.12.1 Offset

Register	Offset
BALUN_TX	28h

A.2.7.12.2 Diagram

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

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

A.2.7.12.3 Fields

Field	Function
31-24 —	Reserved
23-0 RXTX_BAL_TX_CODE	Balun Tuning Cap Settings in Transmit Mode This register contains the Balun Cap Settings for Eight Tuning Ranges. The Ranges and the associated register bits that apply for each Range are as follows : Tuning Range 0 = Radio channel less than or equal to 2375 MHz Tuning Range 1 = Radio channel between 2375 and 2390 MHz Tuning Range 2 = Radio channel between 2390 and 2405 MHz Tuning Range 3 = Radio channel between 2405 and 2420 MHz Tuning Range 4 = Radio channel between 2420 and 2435 MHz Tuning Range 5 = Radio channel between 2435 and 2450 MHz Tuning Range 6 = Radio channel between 2450 and 2465 MHz Tuning Range 7 = Radio channel above 2465 MHz Tuning Range 0 = RXTX_BAL_TX_CODE[2:0] Tuning Range 1 = RXTX_BAL_TX_CODE[5:3] Tuning Range 2 = RXTX_BAL_TX_CODE[8:6] Tuning Range 3 = RXTX_BAL_TX_CODE[11:9] Tuning Range 4 = RXTX_BAL_TX_CODE[14:12] Tuning Range 5 = RXTX_BAL_TX_CODE[17:15] Tuning Range 6 = RXTX_BAL_TX_CODE[20:18] Tuning Range 7 = RXTX_BAL_TX_CODE[23:21]

A.2.7.13 RF Analog Balun RX Mode Control (BALUN_RX)

A.2.7.13.1 Offset

Register	Offset
BALUN_RX	2Ch

A.2.7.13.2 Diagram

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

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

A.2.7.13.3 Fields

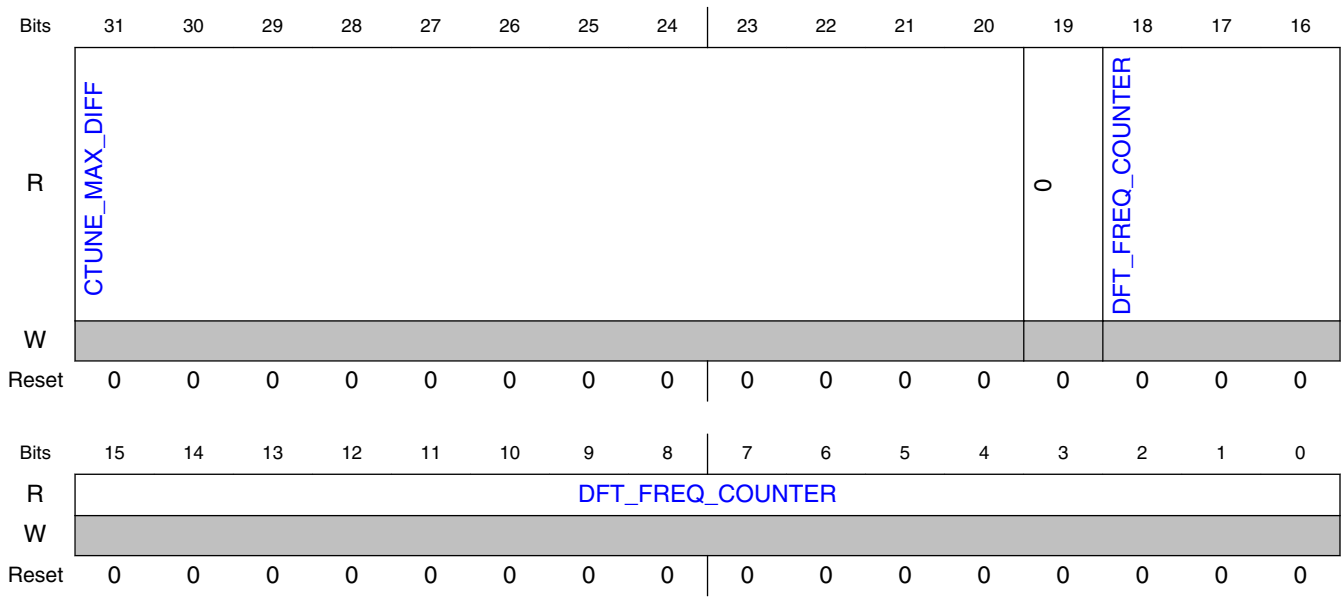
Field	Function
31-24 —	Reserved
23-0 RXTX_BAL_RX_CODE	Balun Tuning Cap Settings in Receive Mode This register contains the Balun Cap Settings for Eight Tuning Ranges. The Ranges and the associated register bits that apply for each Range are as follows : Tuning Range 0 = Radio channel less than or equal to 2375 MHz Tuning Range 1 = Radio channel between 2375 and 2390 MHz Tuning Range 2 = Radio channel between 2390 and 2405 MHz Tuning Range 3 = Radio channel between 2405 and 2420 MHz Tuning Range 4 = Radio channel between 2420 and 2435 MHz Tuning Range 5 = Radio channel between 2435 and 2450 MHz Tuning Range 6 = Radio channel between 2450 and 2465 MHz Tuning Range 7 = Radio channel above 2465 MHz Tuning Range 0 = RXTX_BAL_RX_CODE[2:0] Tuning Range 1 = RXTX_BAL_RX_CODE[5:3] Tuning Range 2 = RXTX_BAL_RX_CODE[8:6] Tuning Range 3 = RXTX_BAL_RX_CODE[11:9] Tuning Range 4 = RXTX_BAL_RX_CODE[14:12] Tuning Range 5 = RXTX_BAL_RX_CODE[17:15] Tuning Range 6 = RXTX_BAL_RX_CODE[20:18] Tuning Range 7 = RXTX_BAL_RX_CODE[23:21]

A.2.7.14 RF Analog DFT Observation Register 1 (DFT_OBSV_1)

A.2.7.14.1 Offset

Register	Offset
DFT_OBSV_1	30h

A.2.7.14.2 Diagram



A.2.7.14.3 Fields

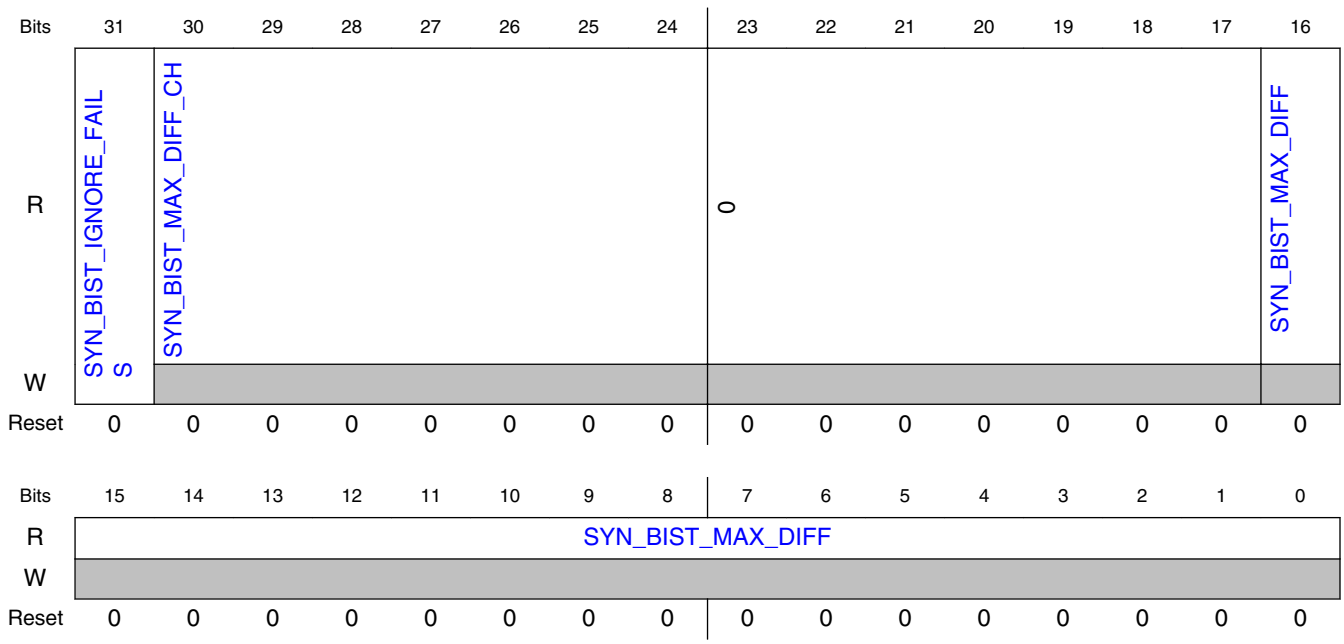
Field	Function
31-20 CTUNE_MAX_DIFF	Maximum Frequency Count Difference found by the Coarse Tune BIST This register is the Absolute Value of the maximum frequency count difference that the Coarse Tune BIST found after tuning to all the Radio channels.
19 —	Reserved
18-0 DFT_FREQ_COUNTER	VCO Frequency Counter Value This Register shows the value of the VCO Frequency Meter after a Count Time has completed. The control bits for the Frequency Meter are in the PLL_DIG register space: FREQ_COUNT_GO, FREQ_COUNT_FINISHED, and FREQ_COUNT_TIME. Note that the counting is done at the VCO frequency which is 2X the Carrier Frequency.

A.2.7.15 RF Analog DFT Observation Register 2 (DFT_OBSV_2)

A.2.7.15.1 Offset

Register	Offset
DFT_OBSV_2	34h

A.2.7.15.2 Diagram



A.2.7.15.3 Fields

Field	Function
31 SYN_BIST_IGNORE_FAILS	PLL Frequency Synthesizer BIST Ignore Fails If this bit is set, then the PLL Frequency Synthesizer BIST will not stop after it is unable to lock on a Radio Channel. Instead the BIST will ignore any channel locking fails and will continue to attempt to lock on every Radio Channel requested
30-24 SYN_BIST_MAX_DIFF_CH	PLL Frequency Synthesizer BIST Worst Channel This register shows the Radio Channel that had the maximum Frequency Meter count difference during the PLL Frequency Synthesizer BIST.
23-17 —	Reserved

Table continues on the next page...

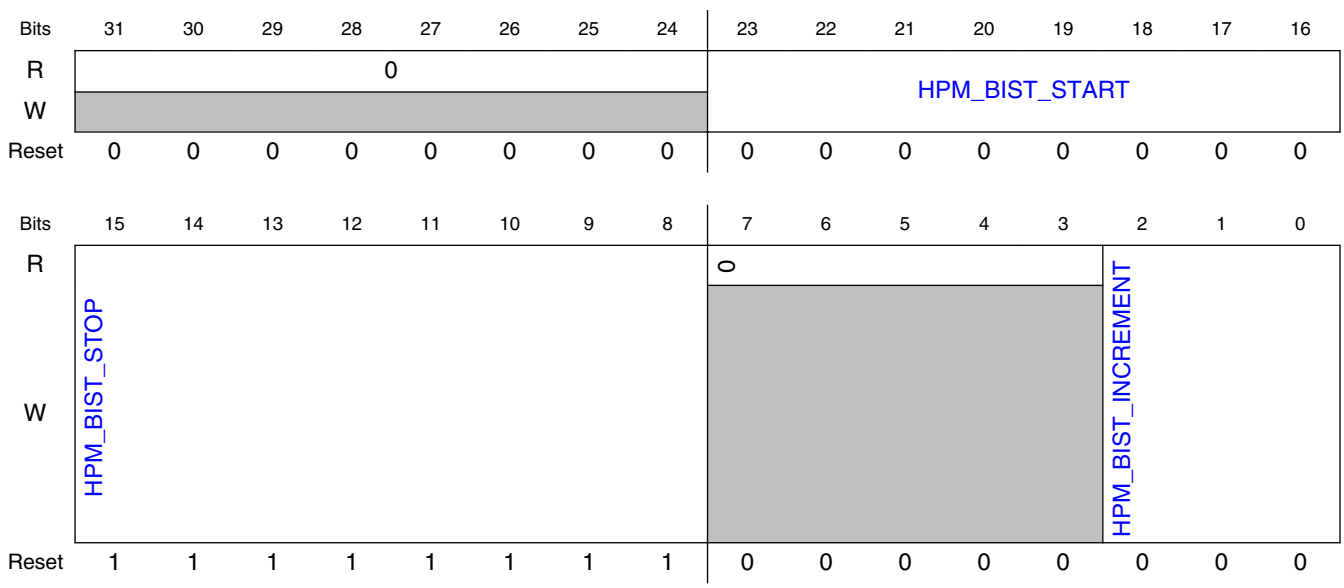
Field	Function
16-0 SYN_BIST_MAX_DIFF	PLL Frequency Synthesizer BIST Worst Frequency Count This register shows the maximum Frequency Meter count difference on a channel during the PLL Frequency Synthesizer BIST.

A.2.7.16 RF Analog DFT Observation Register 3 (DFT_OBSV_3)

A.2.7.16.1 Offset

Register	Offset
DFT_OBSV_3	38h

A.2.7.16.2 Diagram



A.2.7.16.3 Fields

Field	Function
31-24 —	Reserved
23-16	HPM BIST Start Value This is the value at which the HPM BIST will Start

Table continues on the next page...

Field	Function
HPM_BIST_START	
15-8 HPM_BIST_STOP	HPM BIST Stop Value This is the value at which the HPM BIST will Stop
7-3 —	Reserved
2-0 HPM_BIST_INCREMENT	HPM BIST Increment Value This selects the choices for the HPM BIST range or increment. HPM INL BIST Range 0 : dac_range = 8 1 : dac_range = 16 2 : dac_range = 32 3 : dac_range = 64 4 : dac_range = 128 5 : dac_range = 255 6,7 : Reserved HPM DNL BIST Increment 0 : dac_increment = 1 1 : dac_increment = 2 2 : dac_increment = 4 3 : dac_increment = 8 4 : dac_increment = 16 5 : dac_increment = 32 6,7 : Reserved

A.3 Link Layer Memory Map and Register Definition

The Link Layer memory map and description of registers are included in the following register section.

A.3.1 BLE Register Descriptions

BLE base address: 4005_B000h

A.3.1.1 Platform Registers

Platform Register Descriptions for the Bluetooth Link Layer

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	15:11		XX
conn_index[2:0]	10:8	Connection index to specify the command is for which connection engine.	0
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 appendix 1 for details of the registers to be set before setting these instructions.	00

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.	0x0000

				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.	

Field	Bit	Description	Reset
Reserved	15:9	Not used.	XX
Dtm_intr_clr	8	Clear DTM packet reception interrupt. Write to the register with this bit set to 1, clears the interrupt source.	
Reserved	7:6	Not used.	
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
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	15:10	Not used.	XX

Table continues on the next page...

TSM_intr	9	TSM interrupt when BLE protocol is active.	0
Dtm_intr	8	DTM RX packet reception interrupt.	0
Reserved	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 the INIT_STATUS register. This bit is cleared, when firmware clears ALL interrupts by writing to the INIT_INTR_CLEAR register.	0
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	0

Table continues on the next page...

		by writing to the SCAN_INTR_CLEAR register.	
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 enable the interrupts back. This ensures no new interrupts is missed while firmware is processing one.	0x0000

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.	0

Table continues on the next page...

		1 – enable deep sleep mode exit event to interrupt the firmware. 0 – disable deep sleep mode exit interrupt to firmware.	
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
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
Retain_in_dsm2	8	“1” indicates Connection RAM to be retained during DSM2.	0
Reserved	7:5	Reserved for future use.	0
Osc_startup_delay[4:0]	4:0	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 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[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)	00

Wakeup Configuration2 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x278	0x4F0	WAKEUP_CONFIG2	RW	Wakeup configuration to configure the various offsets while waking up from sleep mode or deep sleep mode.	0x00

Field	Bit	Description	Reset
Reserved	15:7	Not used.	XX
Osc_startup_delay[11:5]	6:0	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 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[11:5] is number of slots(625us)`	00

Note: In case of firmware 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	0x0000

				entering sleep mode or deep sleep mode. The threshold values are compared with the inactivity period to determine entry into one of the modes.	

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	0

		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 *_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.	0x0080

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,	0

Table continues on the next page...

Field	Bit	Description	Reset
		interrupt is generated independent of whether exit procedure is initiated by hardware or firmware.	
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.	0

Table continues on the next page...

Field	Bit	Description	Reset
		Base frequency of the sleep_clk input used for generating the internal reference clock of approximate 16Khz frequency.	
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	0

Table continues on the next page...

		the clock gate to shutdown/ wakeup the clock to the module. If 0, the logic has no control and clock is always turned ON.	
Conn_clk_gate_en	3	Connection block clock gate enable. 1 – enable, 0 – disable. 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.	0
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 the operation of the hardware.	0x0000

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
RFU	15:8	Not used	0x0
bb_clk_freq_minus_1	7:3	The frequency information (FREQ – 1) of the clock input to LL Hardware is configured in this register by the firmware during initialization. This information is used inside LL Hardware to derive timing information for the Bluetooth operations. For example, if the frequency of the input clock is 12 MHz this field shall be programmed to 0xB. (1 less than frequency)	0x0
RFU	2:0	Not used	0x0

DSM configuration register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0X51A	0XA34	DSM_CONFIG	RW	DSM configuration register.	0x0008

Field	Bit	Description	Reset
wl_clk_gate_en	15	Whitelist block clock gate enable. 1 = Enable, 0 = Disable. Enables gating of clock to the whitelist 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.	0x0
enable_conn_clk_ctrl[7:0]	[14:7]	Each bit corresponds to one connection engine (bit 7 for conn engine 0). Each of the bits if set to 1 will enable LLH to turn on corresponding CONNECTION engine clock on EXIT_SM command from firmware. To use this feature CON-NECTION NAP modes shall be enabled in LLH.	0x0
enable_init_clk_ctrl	6	If this bit is set to ‘1’ then LLH will turn on INIT engine clock on EXIT_SM command from firmware. To use this feature INIT NAP modes shall be enabled in LLH.	0x0
enable_scan_clk_ctrl	5	If this bit is set to ‘1’ then LLH will turn on SCAN engine clock on EXIT_SM command from firmware. To use this feature SACN NAP mode shall be enabled in LLH.	0x0
enable_adv_clk_ctrl	4	If this bit is set to ‘1’ then LLH will turn on ADV engine clock on EXIT_SM command from firmware. To use this feature ADV NAP mode shall be enabled in LLH.	0x0
enable_eng_clk_ctrl	3	Configuration bit for turning off all engine specific clocks after wakeup from SHUTDOWN (after register restore is complete). If this bit is set to ‘1’ then LLH will turn off all engine specific clocks after	0x1

Table continues on the next page...

		restore_done. To use this feature all NAP modes shall be enabled in LLH.	
restore_done_bypass	2	Configuration bit for bypassing wait for restore_done signal to exit from RESTORE state. If this bit is set to ‘1’ then power control FSM will move to next state without waiting for restore_done from RESTORE state and if this bit is set to ‘0’ then power state machine will wait in RESTORE state till restore_done comes. This bit can be effectively used to reduce wakeup time if restore gets completed in one LP clock cycle.	0x0
store_done_bypass	1	Configuration bit for bypassing wait for store_done signal to enter in to SHUTDOWN. If this bit is set to ‘1’ then power control FSM will move to next state without waiting for store_done from STORE state and if this bit is set to ‘0’ then power state machine will wait in STORE state till store_done comes. This bit can be effectively used to reduce SHUTDOWN entry time if store gets completed in one LP clock cycle.	0x0
dsm_config	0	Configuration bit for DSM SHUTDOWN LLH store and restore control. If this bit is set to ‘1’ then LLH will do the store and restore of its registers on its own during entry and after exit from shutdown. To use this feature DSM SHUTDOWN mode is to be supported and also LLH shall have access to RETENTION RAM.	0x0

Note To enable dynamic power saving though selective clock gating of different engines LL firmware will set the configuration bits corresponding to BT procedures which are active. This will ensure that clocks to only active blocks are enabled during EXIT_SM command from LL firmware to LL hardware. User can over-ride this mode by setting the bits.

Nearest Instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X520	0XA40	NEAREST_INST	RO	This register reports the next immediate instant in terms of bt_clock value (actual value being reported is INSTANT – 3) where hardware has any scheduled BT activity. Firmware reads this register along with the TIM_COUNTER_L register to calculate the possibility of power save mode entry.	0x0000

Field	Bit	Description	Reset
global_nearest_instant	15:0	This field holds the immediate future value of BT_CLOCK where a scheduled Bluetooth activity is supposed to happen. When no Bluetooth activity is running the register will hold current BT_CLOCK value – 3.	0x0000

Auto DSM Configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X274	0X4E8	AUTO_DSM_CONFIG	RW	Auto DSM configuration register.	0x0000

Field	Bit	Description	Reset
Not Used	15:6	Not Used	0x0
Auto_entry_fsm_store_val	5:4	Store/Restore of Auto entry FSM states.	0x0
Dms3_mode_en	3	Enable DSM3 mode for Auto DSM entry feature. 1 – enable, 0 – disable.	0x0

Table continues on the next page...

Dsm2_mode_en	2	Enable DSM2 mode for Auto DSM entry feature. 1 – enable, 0 – disable.	0x0
Fw_wakeup_en	1	Enable wakeup at every FW_WAKEUP_INSTANT by firmware when HW is in sleep mode. 1 – enable, 0 – disable.	0x0
Dsm_auto_sleep_en	0	Reserved	0x0

Firmware wakeup Instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X276	0X4EC	FW_WAKEUP_INSTANT	RW	Wakeup instant register. Program the 16-bit reference clock value for auto-wakeup from deep sleep mode.	0x0000

Field	Bit	Description	Reset
Fw_wakeup_instant	15:0	This field holds the clock instant value when LLH need to exit auto dsm mode.	0x0000

A.3.1.2 Advertising Channel Registers

Advertising Channel Register Descriptions for the Bluetooth Link Layer

##*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 register before issuing start advertise command. The fields in this register correspond to the fields in the LE_Set_Advertising_Parameters HCI command.	0x00E0

Field	Bit	Description	Reset
##peer_rx_txaddr (Read Only)	15	Transmit address field of the received packet indicating the peer address type as public/ random. This field is used by firmware to report peer_addr_type parameter in the connection complete event. 0 – addr type is public. 1 – addr type is random.	0
##peer_addr_resolved_reg (Read Only)	14	Address resolution status of the received packet if the received address was RPA. 0 – RPA resolution fail 1 – RPA resolution pass. Else this bit is set to zero.	0
#slv_index	13:11	Slave index is programmed by FW before initiation procedure for selecting connection engine in slave role connection. Number of supported engines is 8. (Index from 0x0 to 0x7)	0
*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

Table continues on the next page...

		0 - Rxaddr type is public.	
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

Field	Bit	Description	Reset
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
Reserved	0	Reserved for future use.	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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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
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
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 register before issuing start scan command.	0x0010

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

Table continues on the next page...

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	15:7	Not used.	XX
scan_tx_addr[1]	6	MSB of the device's Own address. This bit along with scan_tx_addr[0] decides the own address used for scanner.	0
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 .	0

Table continues on the next page...

		0x00 – Accept advertising packets from any device. A connectable Directed advertising packet not containing the scanner's device address is ignored. 0x01 - Accept advertising packets from only devices in the whitelist . A connectable Directed advertising packet not containing the scanner's device address is ignored. 0x10 - Accept advertising packets from any device. A connectable Directed advertising packet is not ignored if the InitA is a resolvable private address. 0x11 - Accept advertising packets from only devices in the whitelist. A connectable Directed advertising packet is not ignored if the InitA is a resolvable private address. Adv_direct_ind packets which are not addressed to this device are ignored.	
scan_type	2:1	0x00- passive scanning. (default) 0x01- active scanning. 0x10- RFU 0x11- RFU	0
scan_tx_addr[0]	0	Device's own address type. 1 - addr type is random. 0 - addr type is public. Note : Scan_tx_addr[1:0] 0x0 – use public address 0x1 – use random address 0x2 – use RPA if matching entry is available in Resolving list else use public address. 0x3 - use RPA if matching entry is available in Resolving list else use random address.	0

Scan interrupt clear register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
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

Table continues on the next page...

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	0X0000

		Default: 0x0010 (10 ms) Time = N * 0.625 msec Time Range: 2.5 msec to 10.24 sec .	

##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	15:10	Not used.	XX
peer_rpa_resolved_capt (Read only)	9	If received peer address is an RPA, this bit represents the RPA resolution status. ie., 1 – RPA resolution pass. 0 – RPA resolution fail. Else this bit is set to zero.	0
init_tx_addr[1]	8	MSB of the device's Own address type. This bit along with init_tx_addr[0] decides the own address of the device.	0
#Init_conn_index	7:5	Firmware programs the index of the conn engine for the master connection at init start. Index can be in the range 0x0 to 0x7 depending on the number of connections supported.	0x0
Reserved	4	Not used	x
init_filt_policy	3	The Initiator_Filter_Policy is used to determine whether the White List is used or not used.	0

Table continues on the next page...

		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.	
init_rx_addr[1]	2	MSB of the device's peer address type (FW programmed). This bit along with init_rx_addr[0] decides the peer address of the device.	0
rx_addr/rx_tx_addr init_rx_addr[0]	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. LSB of the device's peer address type (FW programmed). Init_rx_addr[1:0] – 0x0 - public address 0x1 - random address 0x2 - public identity address 0x3 - random(static) identity address	0

Tx_addr init_tx_addr[0]	0	Own address type. 1- addr type is random. 0-addr type is public. LSB of the device's Own address type. Init_tx_addr[1:0] - 0x0 – use public address (addr type is zero)	0

		0x1 – use public address (addr type is one) 0x2 – use RPA if matching entry is available in Resolving list else use public address. 0x3 - use RPA if matching entry is available in Resolving list else use random address.	

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 be cleared in a single write operation.	0x0000

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

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
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

#Initiator Anchor Point register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x100	0x200	INIT_ANCHOR_POINT	RW	Firmware programs this register (while programming INIT parameters) with the instant (value of bt_clock) at which the initiator should start the procedure for the first time. This will give firmware a better control in effectively scheduling the new connections in the multiple connection scenarios.	0x0000

Field	Bit	Description	Reset
Init_anchor_point	15:0	The value of bt_clock (625us period) at which initiator should start the procedure.	0

Device Random address lower register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x2C	0x058	DEV_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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x2E	0x05C	DEV_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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x30	0x060	DEV_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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

#Initiator Peer Address Lower Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x66	0x0CC	INIT_PEER_ADDR_L	RW	Lower 16 bit of init peer address. This register is used only if multiple connections are supported.	0x0000

Field	Bit	Description	Reset
Init_peer_addr	15:0	Lower 16 bit of the 48 bit init peer address.	0

#Initiator Peer Address Middle Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x68	0x0D0	INIT_PEER_ADDR_M	RW	Middle 16 bit of init peer address. This register is used only if multiple connections are supported.	0x0000

Field	Bit	Description	Reset
Init_peer_addr	15:0	Middle 16 bit of the 48 bit init peer address.	0

#Initiator Peer Address Upper Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x6A	0x0D4	INIT_PEER_ADDR_H	RW	Lower 16 bit of init peer address. This register is used only if multiple connections are supported.	0x0000

Field	Bit	Description	Reset
Init_peer_addr	15:0	Upper 16 bit of the 48 bit init peer address.	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.

INIT_PEER_ADDR is used only for multiple connection support where peer address for initiation is kept at INIT_PEER_ADDR register and ADV peer address is kept/read from PEER_ADDR register.

White List address type register lower word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xA2	0x144	WL_ADDR_TYPE_L	RW	Stores the address type of the device addresses stored in whitelist	0x0000

Field	Bit	Description	Reset
type	15:0	16 address type bits corresponding to first 16 device address stored in whitelist.	0

		1-Address type is random. 0-Address type is public.	

White List address type register upper word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xA4	0x148	WL_ADDR_TYPE_U	RW	Stores the address type of the device addresses stored in whitelist	0x0000

Field	Bit	Description	Reset
	15:10	Reserved for future use	
type	9:0	10 address type bits corresponding to 17 to 26 device address stored in whitelist. 1-Address type is random. 0-Address type is public.	0

White list enable register lower word

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xA6	0x14C	WL_ENABLE_L	RW	Stores the valid entry bit corresponding to each of the device address stored in the whitelist.	0x0000

Field	Bit	Description	Reset
en	15:0	Stores the valid entry bit corresponding to each of the first 16 device addresses stored in the whitelist memory in the hardware. 1-White list entry is valid. 0-White list entry invalid.	0

White list enable register upper word

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0xA8	0x150	WL_ENABLE_U	RW	Stores the valid entry bit corresponding to each of the device address stored in the whitelist.	0x0000

Field	Bit	Description	Reset
	15:10	Reserved for future use	
en	9:0	Stores the valid entry bit corresponding to each of the 17 to 26 device addresses stored in the whitelist memory in the hardware. 1-White list entry is valid. 0-White list entry invalid.	0

Advertising data transmit FIFO

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 address 16'h74	Data byte 1	Length of scan response host data
	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 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.	0x0000

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

Table continues on the next page...

16-bit	32-bit				
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 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.	0x0006

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 channel configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x27E	n/a	ADVCH_CONFIG	RW	Advertiser channel procedure configuration register.	0x0

Field	Bit	Description	Reset
Reserved	15:1	Not used.	0
Bypass_arb	0	Firmware set this bit when only one procedure (either Advertising, Scanning, Initiation or Connection) is running to indicate hardware that arbitration is not required. 0: Arbitration is required 1: Bypass arbitration	0

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	0xE07F

Field	Bit	Description	Reset
scan_channel_map	15:13	Advertising channels that are enabled for scanning operation.	111

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	Not used.	X
backoff_enable	11	Enable random backoff feature in scanner. 1-enable. 0-disable.	0
Reserved	10:5	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. 15- Enables channel 39 for use. 14- Enables channel 38 for use. 13- Enables channel 37 for use.	0
Reserved	12:8	Not used.	XX

Table continues on the next page...

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
0X462	0x8C4	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], 2nd write - [31:16], 3rd write - [47:32] bits of the device address.	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

Whitelist end address register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X4FE	0x9FC	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 26th 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
0X700	0XE00	CONN_TXMEM_BASE_ADDR	WO	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	0x0000

				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.	

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0X400	0X800	CONN_RXMEM_BASE_ADDR	RO	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 read from Rx memory are read 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,	0x0000

				before initiating connection. After slave connection this value can be obtained by reading this register on the advertiser side.	

Field	Bit	Description	Reset
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 values	0x0000

				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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1E8	0x3D0	CONN_REQ_WORD4	RW	This field defines the 16 bits of the transmit window offset	0x0000

				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.	

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	0x0000

				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.	

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
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. '1' indicates the corresponding data channel is used and '0' indicates the channel is unused.	Data_channels

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

##RPA timer interval register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x582	0xB04	RPA_TIMER_INTERVAL	RW	The RPA timer interval register stores the RPA refresh time interval. (FW programmed).	0x0000

Field	Bit	Description	Reset
rpa_timer_interval_val	15:0	This field holds the value of the RPA refresh time interval in seconds.	0

##Privacy Configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x108	0x210	LL_PRIVACY_CONFIG	RW	This register fields can be set to enable/disable Privacy feature in LLH.	0x0000

Field	Bit	Description	Reset
Reserved	15:2	Reserved for future use.	0
allow_rpa_in_whitelist	1	Allow initiator to receive advertising packets even if peer address resolution fails by adding the received advertiser RPA to the whitelist and to the resolving list as per Errata 6984.	0
privacy_config	0	Whenever this bit is set LLH will try to resolve the address for all ADVCH procedures.	0

##RPA selection index register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x10A	0x214	CURRENT_LOCAL_IRK_INDEX	RW	This register is programmed by firmware before advertising and Initiation procedures. This	0xFF

				bit is used to indicate the local IRK to be used by HW during advertising and Initiation procedures.	

Field	Bit	Description	Reset
Reserved	15:8	Reserved for future use.	0
init_local_irk_index	7:4	This field holds the resolving list index corresponding to the IRK to be used for the INIT procedure. This field is invalid when white list is enabled. Value 0xF – valid IRK for INIT procedure is not set in resolving list.	0xF
adv_local_irk_index	3:0	This field holds the resolving list index corresponding to the IRK to be used for the ADV procedure. This field is also valid when white list is enabled. Value 0xF – valid IRK for ADV procedure is not set in resolving list.	0xF

##Peer identity address type register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x10C	0x218	PEER_IDENTITY_ADDR_TYPE	RW	This register stores the peer identity address type (Public or Random) corresponding to the IRK pair populated in the resolving list. The value is valid if the resolving list entry is valid.	0x0

Field	Bit	Description	Reset
Reserved	15:4	Reserved for future use.	0
peer_id_addr_type	3:0	Peer_id_addr_type [i] = 0/1 => ith resolving list entry corresponds to a peer whose address is public/random.	0

##Resolving list entry valid register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x10E	0x21C	RES_LIST_VALID	RW	This register is used to indicate to HW which resolving list entries are valid/ invalid.	0x0

Field	Bit	Description	Reset
Reserved	15:4	Reserved for future use.	0
device_valid_entry	3:0	Device_valid_entry [i] = 0/1 => ith resolving list entry is invalid/valid.	0

##Advertising parameters 2 register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x0A	0x14	ADV_PARAMS_2	RW	This register stores the own address type for the Advertising procedure.	0x0

Field	Bit	Description	Reset
Reserved	15:2	Reserved for future use.	0
adv_tx_addr	1:0	Device Own_address_type field. 0x0 - Own address type is public address. 0x1 - Own address type is random address. 0x2 - Own address type is RPA. If resolving list contains no matching entry, use public address. 0x3 - Own address type is RPA. If resolving list contains no matching entry, use random address.	0

##Local IRK n registers

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x1100-0x11FE	-	L_IRK_n_x	RW	These registers store the local IRK value for nth resolving list entry. IRK value is of 128 bits and stored in Little Endian format (lowest address register has the LSW).	0x00000000000000000000000000000

Field	Bit	Description	Reset
dev_localirk[n]	127:0	This field has the 128 bit value of local IRK for nth resolving list entry. Note: Refer to appendix 6 for full addressing with respect to each resolving entry.	0

##Peer IRK n registers

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1200-0x12FE	-	P_IRK_n_x	RW	These registers store the peer IRK value for nth resolving list entry. IRK value is of 128 bits and stored in Little Endian format (lowest address register has the LSW).	0x00000000000000000000000000000

Field	Bit	Description	Reset
dev_peerirk[n]	127:0	This field has the 128 bit value of peer IRK for nth resolving list entry. Note: Refer to appendix 6 for full addressing with respect to each resolving entry.	0

##Peer identity address n registers

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1300-0x135E	-	PEER_ID_ADDR_n_x	RW	These registers store the peer identity address for nth resolving list entry. Identity address is of 48 bits and stored in Little Endian format (lowest address register has the LSW).	0x000000000000

Field	Bit	Description	Reset
dev_identity_addr[n]	47:0	This field has the 48 bit value of peer ID address for nth resolving list entry. Note: Refer to appendix 6 for full addressing with respect to each resolving entry.	0

##Local RPA n registers

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x1360-0x13BE	-	L_RPA_n_x	RO	These registers store the local RPA generated by the HW using the local IRK of the nth resolving list entry. RPA is of 48 bits and stored in Little Endian format (lowest address register has the LSW).	0x0000000000

Field	Bit	Description	Reset
self_rpa_list_n	47:0	This field has the 48 bit value of local RPA for nth resolving list entry. Note: Refer to appendix 6 for full addressing with respect to each resolving entry.	0

##Peer RPA n registers

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x13C0-0x141E	0x1CC0 – 0x1CC8	P_RPA_n_x	RO	These registers store the peer RPA generated by the HW using the peer IRK of the nth resolving list entry. RPA is of 48 bits and stored in Little Endian format (lowest address register has the LSW).	0x0000000000

Field	Bit	Description	Reset

Table continues on the next page...

peer_rpa_list_n	47:0	This field has the 48 bit value of peer RPA for nth resolving list entry. Note: Refer to appendix 6 for full addressing with respect to each resolving entry.	0

##RPA timer wrap count register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x586	0xB0C	RPA_TIMER_WRAP_COUNT	RO	The RPA timer wrap count register stores the wrap count value. A wrap is a complete bt_clock roll from 0x0000 to 0xFFFF.	0x0

Field	Bit	Description	Reset
Reserved	**15:10**		**0**
rpa_timer_wrap_count	9:0	This field holds the bt_clock wrap count value.	0

##RPA timer current wrap register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x584	0xB08	RPA_TIMER_CURRENT_WRAP	RO	The RPA timer current wrap register stores the current wrap count value.	0x0

Field	Bit	Description	Reset
Reserved	15:12		0
rpa_timer_current_wrap	11:2	This field holds RPA timer's current wrap count value.	0
rpa_timer_en	1	Indicates whether RPA timer is ON in hardware. 1 – on 0 – off	0
wrap_valid	0	Indicates whether RPA timer's warp is valid. 1 – wrap is valid 0 – wrap is invalid	0

##RPA timer next instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x588	0xB10	RPA_TIMER_NEXT_INSTANT	RO	Holds the next RPA timer instant value.	0x0

Field	Bit	Description	Reset
Reserved	**15:10**		**0**
rpa_timer_next_instant	9:0	Holds the next RPA timer instant value. This instant is valid when wrap valid = 1.	0

##RPA LFSR seed lower register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x9C	0x138	RPA_LFSR_L	RW	Lower seed value of the LFSR for RPA generation.	0x0

Field	Bit	Description	Reset
Rpa_lsfr_l	15:0	Firmware can writes to this register to update the seed value of the LFSR for random RPA generation after the reset.	0

##RPA LFSR seed middle register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x9E	0x13C	RPA_LFSR_M	RW	Middle seed value of the LFSR for RPA generation.	0x0

Field	Bit	Description	Reset
Rpa_lsfr_m	15:0	Firmware can writes to this register to update the seed value of the LFSR for random RPA generation after the reset.	0

##RPA LFSR seed higher register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xA0	0x140	RPA_LFSR_H	RW	Upper seed value of the LFSR for RPA generation.	0x0

Field	Bit	Description	Reset
Reserved	15:12	RFU	**0**
Rpa_lsfr_u	11:0	Firmware can writes to this register to update the seed value of the LFSR for random RPA generation after the reset.	0

##Privacy Mode register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x280 – 0x28E	0x500 – 0x51C	PRIVACY_MODE_0 – PRIVACY_MODE_7	RW	Mode of the privacy address added for each entry in the resolving list.	0x0

Field	Bit	Description	Reset
Reserved	15:1	RFU	**0**
Privacy_mode	0	This bit defines the mode of the device is network privacy or device privacy. 0 – Network Privacy Mode 1 – Device Privacy Mode	0

A.3.1.3 Data Channel Registers

Data Channel Register Descriptions for the Bluetooth Link Layer

Transmit window offset register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x40	0x080	TRANSMIT_WINDOW_OFFSET	RW	Stores the transmit window offset 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_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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
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
Reserved	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
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	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
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 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 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. After firmware reads a packet, it decrements the counter by issuing the PACKET_RECEIVED command from the commander.	0
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.	0x631F

Field	Bit	Description	Reset

Table continues on the next page...

pdu_index_auto_updt	**15**	This bit enable the LL hardware to update the index of the PDU being queued in the data buffer. 0x0- LL Hardware to update the index of the PDU being queued. 0x1- LL firmware to update the index when PDU is queued.	0x0
mask_suto_at_update	14	This bit is used to enable/ disable masking of internal hardware supervision timeout trigger when switching from old connection parameters to new parameters. 1 - Enable 0 - Disable	0x01
extend_cu_tx_win	13	This bit is used to enable/ disable extending the additional rx window on slave side during connection update in event of packet miss at the update instant. 1 - Enable 0 - Disable	0x01
slv_md_config	12	Configuration is provided to send last data packet with MD bit set/zero or one (only for slave mode). SLV_MD_CONFIG bit has effect only when md_bit_ctr bit is not set 0 – The MD bit is set based on the transmit buffer empty condition. The last packet goes with a MD bit set since the FIFO would not be empty as we have to retain the data in the buffer till we realize the ACK from the remote side. 1 - MD bit will be controlled based on the availability of next data in the FIFO. So send last data packet with MD bit zero or one. If the data transmitted is the last packet in the FIFO the MD bit will be '0' in that packet. Only if the next location holds data then MD bit will be '1'.	0x0

Table continues on the next page...

aww_en	11	This field indicates whether window widen optimization is enable or not. 0 – It indicates AWW is enable 1 – It indicates AWW is disable	0x0
sl_dsm_en	10	This resiter field indicates whether slave latency dsm is enable or not. 0 – It indicates SL_DSM is enable 1 – It indicates SL_DSM is disable	0x0

index_not_in_addr	9	This register field indicates whether connection index is present in address or not. 1 – Index is not in address 0 – Index is in address	0x1
Sw_cntrl_md	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

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 LLH 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, LLH 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_expired_intr	15	If this bit is set ping timer nearly expired interrupt is enabled.	0
*ping_timer_expired_intr	14	If this bit is set ping timer expired interrupt is enabled.	0
Reserved	13:10	Unused.	0
rx_bad_pdu_int_en	9	If this bit is set packet receive bad pdu interrupt is enabled.	0

Table continues on the next page...

		Effective only when bit 6 is set.	
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

Table continues on the next page...

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 conn_config 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 window_widen_intvl register value to calculate the window widening size.	00A

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. Firmware shall update this register with proper index before writing/ reading the connection-specific registers (refer to register summary before for the connection specific register set). This register is relevant only for multiple connection support. If multiple connection is supported but number of connections is made one then the register value needs to be kept 0x0.	0x0000

Field	Bit	Description	Reset
RFU	15:3	Reserved	0
fw_conn_index	2:0	This field is used to index the multiple connections existing. Range is 0 to one less than maximum number of connections supported (limited to 4 as of now). 0x0 – Connection Engine 0 in LLH 0x1 – Connection Engine 1 in LLH	0

Note – An alternative for programming of the conn index register for every access to the connection registers is to pass the conn index along with the LLH address itself which requires address space to be available for this mapping.

*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 connection. To clear interrupt for a specific	0x0000

				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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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
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

Table continues on the next page...

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. This shall be a multiple of 1.25 ms in the range of 7.5 ms to 4.0 s.	0x0000

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

Table continues on the next page...

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	RO	This is the free running counter, connEventCounter as defined by Bluetooth spec.	0x0000

Field	Bit	Description	Reset
connectionEventCounter	15:0	Firmware will read the instantaneous Event counter from this register, during connection update and channel map update 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.	0x0000

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

Table continues on the next page...

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. Pause the data transfer on the connection. The current_pdu_index in hardware does not move to next index until pause_data is cleared.	0
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.	0

		Hardware will start the next data transmission from the index indicated by this field.	

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_new[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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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_latency_newl[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_so_to_new[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
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

Table continues on the next page...

sl_conn_interval_new[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 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.	0x000A

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	0x0000

				the time to get the response for the control procedures for which timeout rules are specified in the specification.	

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_resp_timer_expire	15:0	This field defines the clock instant at which the next PDU response timeout event will occur on a connection.	0x0000

		This is with reference to the 16-bit internal reference clock.	

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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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	15:9	Unused	0
Set/Clear	8	Used to set the SENT bit in hardware for the selected packet buffer. 1 – packet queued	0

Table continues on the next page...

		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. Note: This register has a different meaning in the Read-path	
List_index	7:0	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	15:9	Unused	0x0000

Table continues on the next page...

Set/clear	8	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. 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	0
List_index	7:0	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

Table continues on the next page...

Reserved	15:8	Unused	0x0000
tx_sent	7:0	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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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	15:8	Unused	0x0000
tx_ack	7:0	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.	0

		Note : This register has a different meaning in the Write-path.	

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.

Device Data List Status Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x290	0x520	DEVICE_DATA_LIST_STATUS	RO	Status of data packets queued and sent.	0x0000

Field	Bit	Description	Reset
Reserved	15:8	Unused	0x0000
device_data_list_status	7:0	These bits the status of data pecked queued and sent at device level. If the particular bit is set to ‘1’ indicates that there either pending PDU in the queue or pending ACK to be cleared. Based on the status of this register FW will queue next PDU. Note : This register is used only in shared connection tx_fifo.	0

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
0x200 - 0x21E	0x400- 0x43C	DATA_MEM_DESCRIPTOR0 - DATA_MEM_DESCRIPTOR15	RW	Descriptor for packet stored in the each of the transmit buffer which includes the packet specific information like length and LLID.	0x0000

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.	0x000A

Field	Bit	Description	Reset
Reserved	7:2	RFU	0
llf_sl_dsm_ww_ctrl	3	Control bit to enable or disable the wakeup delay when slave latency is enabled.	1
slave_latency_en_sts	2	Read Only . Firmware reads this bit to know the status of slave latency enable.	0
llf_sl_dsm_en	1	LLF assisted slave latency dsm feature is enabled by setting this bit. When slave latency is enabled, this feature enables the slave to awaken out of DSM when it is necessary for the slave to receive a packet.	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

Adaptive Window Widening Config

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x50A	0xA14	AWW_CONFIG	RW	Enabling/Disabling of various AWW specific functions and AWW configurable parameters.	0xE244

Field	Bit	Description	Reset
aww_error_tolerence_en	15	This bit enables/disables the access code bit error tolerance functionality. 1 – enable 0 - disable	1
aww_tolerence_threshold	14:10	This value sets the lower bound on tolerance of bit errors in access code to overlook AWW history recapture. Eg., 0x18 implies 32-24 = 8 bit errors are tolerable.	0x18
aww_alg_auto_switch_en	9:9	This bit is used to enable/ disable the functionality of automatic switching between the AWW algorithms i.e The Conservative Mode and Greedy Mode. 1 – Enable 0 – Disable	1
aww_alg_sel	8:8	IF the AWW algorithm auto switch functionality is disabled, this bit is used to manually select the AWW algorithm to be used. 1 – Greedy Mode 0 – Conservative Mode	0
constancy_cnt_init_val	7:4	This value defines the number of connection events for which the adaptive window module checks if master anchor points variations fall within set threshold before switching from conservative mode to greedy mode. Valid Range : (0x1,0xf)	0x04
Obs_ce_cnt_init_val	3:0	This value defines the number of connection events for which the adaptive window module performs measurements of master anchor point swing before it applies the optimized window in accordance with conservative mode aww algorithm. Valid Range : (0x1,0xf)	0x04

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 calculated slave window widening value to be used as final window widen value.	0x10

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. Unit is 1.25ms	0x0000

Cumulative hop increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x25A	0x4B4	SL_CONN_HOP_INCREMENT	RW	This register holds the value of cumulative hop increment corresponding to the Slave latency.	0x00

Field	Bit	Description	Reset
Sl_conn_hop_increment	5:0	Firmware writes the cumulative hop increment value corresponding to the slave latency upon connection creation complete before enabling Slave latency. Range of the value is 0 to 36.	0x00

Cumulative window widen register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x25C	0x4B8	SL_CONN_WINDOW_WIDEN	RW	This register holds the value of cumulative window widen value corresponding to the slave latency.	0x0000

Field	Bit	Description	Reset
Sl_conn_window_widen	15:0	Firmware writes the cumulative window widen value corresponding to the slave latency upon connection creation complete before enabling Slave latency. Unit is in microseconds.	0x0000

Connection Update New slave latency hop increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x25E	0x4BC	CONN_UPDATE_NEW_SL_HOP_INCREMENT	RW	The (slaveLatency * hopIncrement) value that will be effective after the connection update instant.	0x00

Field	Bit	Description	Reset
Sl_hop_increment_new[5:0]	5:0	This register will have the new SL_New * hopIncrement value that the hardware will use after the connection update instant. Before the instant, the hop increment in the register SL_CONN_HOP_INCREMENT will be used by hardware. Range of the value is 0 to 36.	0x00

Connection Update New slave latency window widen register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x260	0x4C0	CONN_UPDATE_NEW_SL_WIN_WIDEN	RW	The cumulative window widen value corresponding to the new slave latency after the update instant.	0x0000

Field	Bit	Description	Reset
Sl_window_widen_new[15:0]	15:0	The cumulative window widen value after the connection update instant. Before the instant, the window widen in the register SL_CONN_WINDOW_WIDEN will be used by hardware. Unit is in microseconds.	0x0000

Previous connection instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x262	0x4C4	PREVIOUS_CONN_INSTANT	RO	Reference BT clock value of the previous anchor point.	0x0000

Field	Bit	Description	Reset
Previous_conn_instant	15:0	Read only register. Holds the value of the reference BT clock value at the previous connection event that LLH participated.	0x0000

Previous Connection instant event count

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x26E	0x4DC	PREVIOUS_CONN_CE_COUNT	RO	Connection event count of the previous anchor point.	0x0000

Field	Bit	Description	Reset
previous_conn_ce_count	15:0	Read only register. Holds the value of the event counter at the previous connection event that LLH participated.	0x0000

Previous Connection channel

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x270	0x4E0	PREVIOUS_CONN_CH	RO	Connection channel of the previous anchor point.	0x0000

Field	Bit	Description	Reset
previous_conn_ch	15:0	Read only register. Holds the value of the channel at the previous connection event that LLH participated.	0x0000

Slave latency connection event counter increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x264	0x4C8	CE_COUNT_INCREMENT_N	RW	This register holds the value of connection event count N.	0x0000

Field	Bit	Description	Reset

Table continues on the next page...

Ce_count_increment_n	15:0	Firmware programs the value of connection event count N corresponding to the elapsed time (from the previous anchor point) and expected next anchor point (anchor point when it wants to exit slave latency).	0x0000

Slave latency connection interval increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x266	0x4CC	CONN_INTERVAL_INCREMENT_N	RW	This register holds the value of cumulative connection interval for the connection event count N.	0x0000

Field	Bit	Description	Reset
Conn_interval_increment_n	15:0	Firmware writes the cumulative connection interval for the connection event count N corresponding to the elapsed time (from the previous anchor point) and expected next anchor point (anchor point when it wants to exit slave latency). Unit is 1.25ms	0x0000

Slave latency hop increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x268	0x4D0	CONN_HOP_INCREMENT_N	RW	This register holds the value of cumulative hop increment for the connection event count N.	0x00

Field	Bit	Description	Reset
Conn_hop_increment_n	5:0	Firmware writes the cumulative hop increment for the connection event count N corresponding to the elapsed time (from the previous	0x00

		anchor point) and expected next anchor point (anchor point when it wants to exit slave latency). Range of the value is 0 to 36.	

Slave latency window widen increment register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x26A	0x4D4	CONN_WINDOW_WIDEN_N	RW	This register holds the value of cumulative window widen for the connection event count N.	0x0000

Field	Bit	Description	Reset
Conn_window_widen_n	15:0	Firmware writes the cumulative window widen for the connection event count N corresponding to the elapsed time (from the previous anchor point) and expected next anchor point (anchor point when it wants to exit slave latency). Unit is in microseconds.	0x0000

Slave latency new connection instant register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x26C	0x4D8	ANCHOR_POINT_INCREMENT_N	RW	This register holds the value of the distance for connection event count N corresponding to the elapsed time.	0x0000

Field	Bit	Description	Reset
Anchor_point_increment_n	15:0	Firmware write the value of the distance for the connection event count N corresponding to the elapsed time from the previous anchor point. It calculate the distance as	0x0000

		(N * CI) – (((SL_CONN_WINDOW_WIDEN_N) slots) + 1)). Unit is 0.625ms LLH use this value to check the validity the instant passed for the new derived anchor point with reference to the PREVIOUS_INSTANT.	

**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).	0x0000

				This together with CONN_PING_TIMER_NEXT_EXP_WRAP will provide the correct status of ping timer duration.	

**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
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

#Connection Arbiter Parameter register

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x102	0x204	CONN_ARB_PARAMS	RO	Register holds the highest priority request to the connection arbiter from the connection engines.	0xF

Field	Bit	Description	Reset
Reserved	15:4	Not used	--
priority_pointer	3:0	Pointer for the highest priority request to the connection arbiter.	0xF

Conn_SUTO_CI_Ratio register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x104	0x208	CONN_SUTO_CI_RATIO	RW	Firmware programs this register with connection supervision timeout to connection interval ratio at the time of connection creation/ connection update.	0x0000

Field	Bit	Description	Reset
RFU	15:13	Not used	0x0
conn_suto_ci_ratio[12:0]	12:0	This register will have the ratio of the connection supervision timeout to the connection interval as programmed by the firmware before a connection creation or a connection update. The value is used by hardware to update the priority of a connection link that is nearing connection timeout due to packet miss.	0x0000

Connection Priority Config register

Addr	Addr 32-bit	Register Name	RW	Description	Reset

Table continues on the next page...

0x106	0x20C	CONN_PRIORITY_CNFG	RW	This register is used by the firmware to override the hardware priority and set the priority of each connection link in the connection arbiter. The value of Zero has the least priority.	0x0000

Field	Bit	Description	Reset
priority_sel	15	If this bit is set to '1' connection arbiter will use the firmware programmed priority set in this register for each connection engine.	0
RFU	14:12	Not used	0x0
fw_priority3	11:9	Firmware programmed priority for connection engine 3.	0x0
fw_priority2	8:6	Firmware programmed priority for connection engine 2.	0x0
fw_priority1	5:3	Firmware programmed priority for connection engine 1.	0x0
fw_priority0	2:0	Firmware programmed priority for connection engine 0.	0x0

Connection Update New SUP_TO to CI ratio

Addr	Addr 32-bit	Register Name	RW	Description	Reset
0x1DA	0x3B4	CONN_UPDT_NEW_SU_CI_RATIO	RW	The Supervision Timeout to conn interval ratio that will be effective after the connection update instant. Firmware programs this register along with other connection update parameter registers.	0x0000

Field	Bit	Description	Reset
RFU	15:13	Not used.	0x0
conn_suto_ci_ratio_new	12:0	This register will have the new supervision timeout to connection interval ratio that the hardware will use after the connection update instant. Before the instant, the value in the register CONN_SUTO_CI_RATIO will be used by hardware.	0x0000

#Slave Window Offset Full register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset	
0x1DC	0x3B8	SL_WIN_OFF_FULL	RW	This register is programmed with the maximum possible window widen value by the firmware when dsm entry during slave latency feature is enabled. This value is used by the hardware to calculate the next ce instant being reported to firmware.	0x0000	

Field	Bit	Description	Reset
RFU	15:6	Not used.	0x0
sl_window_off_full	5:0	Value of maximum possible window widen programmed by the firmware when slave latency dsm feature is enabled.	0x0000

Connection RSSI register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset	
0x518	0xA30	CONN_RX_RSSI	RO	This connection register holds the RSSI value of the last good (empty/non-empty) packet received by the specific connection.	0x0000	

Field	Bit	Description	Reset

Table continues on the next page...

conn_rx_rssi	15:0	RSSI during the last good packet received in the specific connection.	0x0000

Connection Rx Memory read enable register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset	
0x522	0xA44	LLH_MEM_READ_ENABLE_CONTROL_REG	WO	This register is used by the firmware to control read access to the connection Rx memory.	0x0000	

Field	Bit	Description	Reset
Reserved	15:14	Not Used	0x0
conn_rxmem_rd_ctrl	0	When this bit is '1', connection Rx memory is enabled.	0

Tx memory configuration register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x27A	0x4F4	TXMEM_CONFIG	RW	This register fields can be set to configure the size of the data PDUs.	0x700

Field	Bit	Description	Reset
RFU	15:12	Reserved	0x0
Number_of_packets_minus_1	11:9	Maximum Number of Data PDUs in Tx FIFO.	0x03
Max_data_pdu_size	8:0	Maximum data packet length.	0x100

Tx memory configuration register 2

Addr	Addr	Register Name	RW	Description	Reset

Table continues on the next page...

16-bit	32-bit				
0x27C	0x4F8	TXMEM_CONFIG_2	RW	This register fields can be set to configure the base address of the Control PDU.	0x300

Field	Bit	Description	Reset
RFU	15:11	Reserved	0x0
Max_total_data_size	10:0	Total size of Data PDUs. i.e. Maximum Number of Data PDUs * Maximum Data Packet Length as configured in TXMEM_CONFIG register. It is base address of first Control PDU.	0x300

A.3.1.4 Test and Debug Registers

Test and Debug Register Descriptions for the Bluetooth Link Layer

DTM control register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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
Reserved	15:10	Reserved for future use	0
dtm_conttx_en	13	Write: 1 – To enable continuous transmit mode 0 – To disable continuous transmit mode	0
Reserved	12:10	Not Used	
Pkt_payload[2:0]	9:7	Payload type as per the HCI parameter. 0x00 Pseudo-Random bit sequence 9	0

Table continues on the next page...

		0x01 Pattern of alternating bits ‘11110000 0x02 Pattern of alternating bits ‘10101010’ 0x03 Pseudo-Random bit sequence 15 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. Read: 1 – Indicates DTM test ON 0 – Indicates DTM test OFF Write: 1 – To enable continuous receive mode 0 – To disable continuous receive mode	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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

Table continues on the next page...

		0 – no change from normal behavior	
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
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.	0

Table continues on the next page...

		1 – corrupt CRC 0 – no change	
Rx_crc_err	2	Receive packet CRC error indication. 1 – Indicate CRC error of received packet, irrespective of good/bad CRC 0 – No change from normal behavior	0
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

DTM transmit packet count register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xC2	0x184	DTM_TX_PKT_COUNT	RO	Count of the number of LE packets transmitted when device is configured in transmit test mode.	0x0000

Field	Bit	Description	Reset
tx_packet_count[15:0]	15:0	Number of packets transmitted in transmit test mode. 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
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

DTM Length register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0xCE	0x19C	LE_DTM_TEST_LEN	RW	LE Direct Test Mode (DTM) length register. Used to configure the payload length of direct test mode (DTM) operation.	0x0000

Field	Bit	Description	Reset
Reserved	15:8	Reserved for future use	0
dtm_len	7:0	0x00-0xFF (except 0x3F) Length in bytes of payload data in each packet 0x3F – Continuous Transmit mode.	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 Decimal	Output Sleep Frequency(KHz)	PPM	SCA	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.

DTM 2 wire UART *Baud rate configuration*

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x580	0xB00	DTM_2WIRE_CONFIG	RW	This value will go to different baud rate generator module for DTM 2 wire UART.	0x006F

Field	Bit	Description	Reset
dtm_2wire_baud_config	15:0	This value will be given to baud clock generator module. For different input clock frequency to LLH, this value will be different to generate corresponding baud clock as mentioned in below table.	0x006F

Radio clock	Baud rate	Configuration Value
26 MHz	1200	0x2A3B
	2400	0X151E

Table continues on the next page...

	9600	0X0546
	14400	0X0384
	19200	0X02A1
	38400	0X0152
	57600	0X00E1
	115200	0X006F
64 MHz	1200	0X19FA
	2400	0X0CFE
	9600	0X033E
	14400	0X0229
	19200	0X019E
	38400	0X00D0
	57600	0X008A
	115200	0X0042

Note: For lec_top (LLH + PHY) environment input frequency is 64 MHz. Value mentioned on above table for 64 MHz is w.r.t 16 MHz because we divide 64 MHz value by 4 internally. This 16 MHz derived value is given as an input to LLH and used for further calculation.

DTM Offset Configuration Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x118	0x230	DTM_OFFSET_CONFIG	RW	Contains configurable parameters for DTM function enhancement request.	0x012D

Field	Bit	Description	Reset
Reserved	15:11	Reserved for future use.	
dtm_offset_us_buffer	10:1	Represents TIFS value ‘150’ in the calculation of dtm offset.	0x96
dtm_offset_en	0	If set to 1, DTM offset enhancement is enabled. If set to 0, DTM offset enhancement is disabled.	0x1

-

DTM Transmit Packet Configuration Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x11A	0x234	DTM_TX_PKT_CONFIG	RW	Configures number of DTM packets to be transmitted when DTM Tx operation is triggered and automatically stops the DTM Tx operation after that many number of packets are transmitted.	0x0000

Field	Bit	Description	Reset
dtm_tx_pkt_config[15:0]	15:0	If this field value is non-zero, it represents number of DTM packets to be transmitted when dtm_start command is received. DTM Tx operation is automatically stopped when DTM Tx packet count equals to the configured value. If this field value is zero, DTM continues to transmit packets till dtm_stop command is issued.	0

DTM Expected Receive Packet Length Register

-

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x11C	0x238	DTM_EXP_RX_LEN	RW	Contains configurable parameters for DTM timeout feature.	0x0000

Field	Bit	Description	Reset
Reserved	15:9	Reserved for future use.	
dtm_exp_rx_length	8:1	Configured length value for DTM receiver.	0x00
dtm_exp_rx_len_en	0	If set to 1, DTM timeout feature is enabled. If set to 0, DTM timeout feature is disabled.	0x0

Packet Payload Status Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x11E	-	PKT_PAYLOAD_STATUS	RO	Contains CRC status and Payload length value of received packet.	0x0000

Field	Bit	Description	Reset
Reserved	15:9	Reserved for future use.	
crc_status	8	Contains the value of CRC status of the received packet. Updates each time when packet reception is over.	0x0
payload_length	7:0	Contains the value of payload length field. Updated each time when packet reception is over.	0x00

Packet RSSI Status Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x120	-	PKT_RSSI_STATUS	RO	Contains RSSI value for the received packet.	0x000

Field	Bit	Description	Reset
pkt_rssi_status	15:0	Contains the RSSI value of the received packet. Updated each time when packet reception is over.	0x0000

A.3.1.5 Interface Registers

Interface Register Descriptions for the Bluetooth Link Layer

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	0x0000

				for access address match, the access match trigger 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

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

RSSI Register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
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

RF_ACTIVE_PERIOD register

Addr 16-bit	Addr 32-bit	Register Name	RW	Description	Reset
0x5A	0x0B4	rf_active_period	RW	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: 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_polarity	6	Polarity bit for rf_active	

Table continues on the next page...

		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

A.3.1.6 Instruction Set

Instruction Set Description for the Bluetooth Link Layer

Command	Opcode	Description
START_ADV	0x40	Start Advertiser operation. The associated Advertiser configuration registers are programmed before the command is issued.
STOP_ADV	0x41	Stop advertiser operation.
START_SCAN	0x42	Start scanner operation. The associated configuration registers must be programmed before the command is issued.
STOP_SCAN	0x43	Stop the scanner operation.
START_INIT	0x44	Start connection creation operation. The associated configuration registers must be programmed before the command is issued.
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.
DTM_RX_START	0x47	Start Direct Test Mode Receive Test. The associated configuration registers must be programmed before the command is issued.
DTM_STOP	0x48	Stop Direct Test Mode.
UPDATE_CHAN_MAP	0x4B	Update channel map for the connection.

Table continues on the next page...

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.
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.
WL_CLK_ON	0x5d	Turn on clock to whitelist block
WL_CLK_OFF	0x5e	Turn off clock to whitelist block
UPDATE_CONN	0x68	Update connection parameters. Deprecated.
RC_ANCHOR_POINT	0x69	Instruct LLH to check the validity of the set and recalculate the new connection anchor point parameter if valid.
ENTER_AUTO_DSM	0x6A	Indicates CPU is idle.
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
ENTER_DSM_SHUTDOWN	0x77	Enter deep sleep mode with shutdown
STORE_START	0x78	Firmware triggers the store process
RESTORE_START	0x79	Firmware triggers the restore process
DATA_RESTORE_ON	0x7A	Firmware data restore enable
DATA_RESTORE_OFF	0x7B	Firmware data restore disable
START_RPA_TIMER	0x7C	Start Privacy RPA timer
STOP_RPA_TIMER	0x7D	Stop Privacy RPA timer
START_RPA_GEN	0x7E	Start RPA address generation
SOFT_RESET	0x80	Software reset. Resets all the hardware registers (except a few registers related to radio initialization).

Table continues on the next page...

LOAD_RAND_NUM	0x8C	Load random seed to LFSR for RPA generation

A.3.2 BTLE_RF register descriptions

A.3.2.1 BTLE_RF Memory map

BTLE_RF base address: 4005_B000h

Offset	Register	Width (In bits)	Access	Reset value
600h	BLUETOOTH LOW ENERGY PART ID (BLE_PART_ID)	16	RO	0005h
604h	BLE DSM STATUS (DSM_STATUS)	16	RO	See description.
608h	BLE MISCELLANEOUS CONTROL (MISC_CTRL)	16	RW	0000h
60Ch	BLE STATE MACHINE STATUS (BLE_FSM)	16	RO	0000h

A.3.2.2 BLUETOOTH LOW ENERGY PART ID (BLE_PART_ID)

A.3.2.2.1 Offset

Register	Offset
BLE_PART_ID	600h

A.3.2.2.2 Diagram

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

A.3.2.2.3 Fields

Field	Function
15-0 BLE_PART_ID	BLE Part ID 0000000000000000b - Pre-production 0000000000000001b - Pre-production 0000000000000010b - KW40 0000000000000011b - KW41 0000000000000100b - K3S 0000000000000101b - KW35/KW36

A.3.2.3 BLE DSM STATUS (DSM_STATUS)

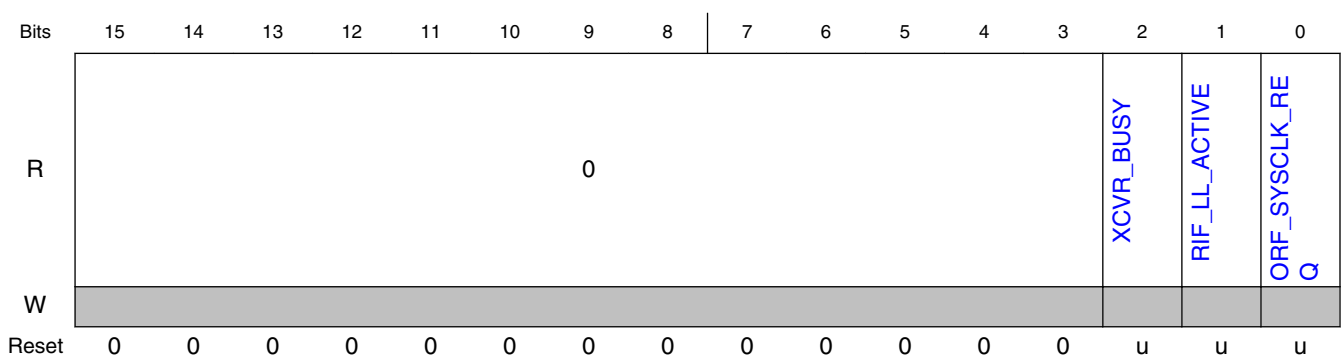
A.3.2.3.1 Offset

Register	Offset
DSM_STATUS	604h

A.3.2.3.2 Function

BLE Deep Sleep Mode Status Register

A.3.2.3.3 Diagram



A.3.2.3.4 Fields

Field	Function
15-3 —	Reserved.

Table continues on the next page...

Field	Function
2 XCVR_BUSY	Transceiver Busy Status Bit 0b - RF Channel in available (TSM is idle) 1b - RF Channel in use (TSM is busy)
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).

A.3.2.4 BLE MISCELLANEOUS CONTROL (MISC_CTRL)

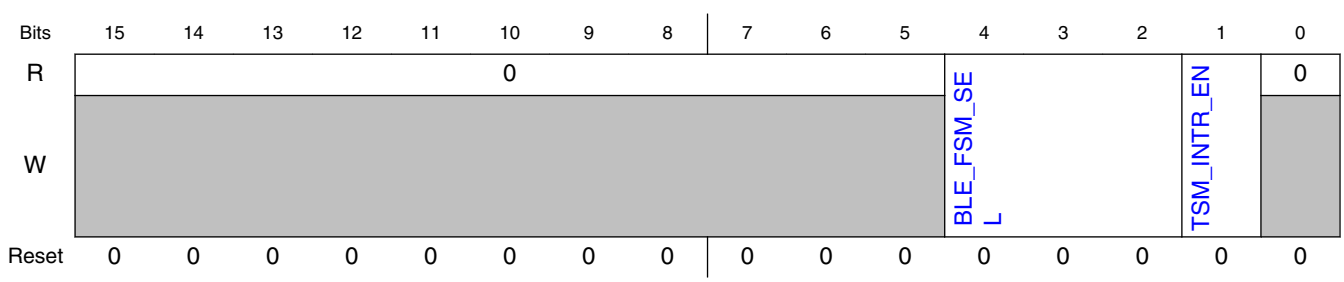
A.3.2.4.1 Offset

Register	Offset
MISC_CTRL	608h

A.3.2.4.2 Function

BLE Miscellaneous Control Register

A.3.2.4.3 Diagram



A.3.2.4.4 Fields

Field	Function
15-5 —	Reserved.

Table continues on the next page...

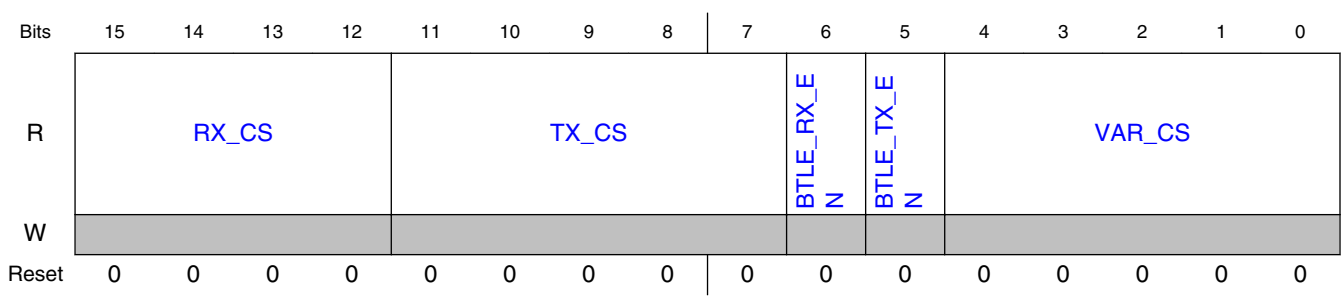
Field	Function
4-2 BLE_FSM_SEL	BLE FSM Selector BLE_FSM[2:0] selects which BLE state machine appears in the VAR_CS field of the BLE_FSM register, according to the following table: 0: VAR_CS = {2'b00, dtm_cs[2:0]} (DTM Current State) 1: VAR_CS = { adv_cs[4:0]} (ADV Current State) 2: VAR_CS = { scan_cs[4:0]} (SCAN Curernt State) 3: VAR_CS = { init_cs[4:0]} (INIT Current State) 4: VAR_CS = {1'b0, connection_cs[3:0]} (CONN Current State, using Connection Engine 0) 5: Reserved 6: Reserved 7: Reserved
1 TSM_INTR_EN	TSM Interrupt Enable This control bit enables the TSM Interrupt for BLE. If TSM_INTR_EN=1 and a TSM Interrupt occurs during a BLE TX or RX operation, bit [9] of the BLE EVENT_STATUS register will become set.
0 —	Reserved.

A.3.2.5 BLE STATE MACHINE STATUS (BLE_FSM)

A.3.2.5.1 Offset

Register	Offset
BLE_FSM	60Ch

A.3.2.5.2 Diagram



A.3.2.5.3 Fields

Field	Function
15-12	BLE RX State Machine Current State

Table continues on the next page...

Field	Function
RX_CS	Current State of *rx_cs[3:0]* (BLE RX state machine)
11-7 TX_CS	BLE TX State Machine Current State Current State of *tx_cs[4:0]* (BLE TX state machine)
6 BTLE_RX_EN	BLE RX Enable Control to TSM Current State of *btle_rx_en* (BLE TSM RX request input)
5 BTLE_TX_EN	BLE TX Enable Control to TSM Current State of *btle_tx_en* (BLE TSM TX request input)
4-0 VAR_CS	Variable State Machine Current State One of several BLE state machines can be mapped to VAR_CS[4:0]. See description of BLE_FSM_SEL[2:0] in the MISC_CTRL register

A.3.3 GENERIC FSK register descriptions

A.3.3.1 FSK Memory map

GENFSK base address: 4005_F000h

Offset	Register	Width (In bits)	Access	Reset value
0h	IRQ CONTROL (IRQ_CTRL)	32	RW	See description.
4h	EVENT TIMER (EVENT_TMR)	32	RW	See description.
8h	T1 COMPARE (T1_CMP)	32	RW	00FF_FFFFh
Ch	T2 COMPARE (T2_CMP)	32	RW	00FF_FFFFh
10h	TIMESTAMP (TIMESTAMP)	32	RO	See description.
14h	TRANSCEIVER CONTROL (XCVR_CTRL)	32	RW	See description.
18h	TRANSCEIVER STATUS (XCVR_STS)	32	RO	See description.
1Ch	TRANSCEIVER CONFIGURATION (XCVR_CFG)	32	RW	See description.
20h	CHANNEL NUMBER (CHANNEL_NUM)	32	RW	0000_0000h
24h	TRANSMIT POWER (TX_POWER)	32	RW	0000_0000h
28h	NETWORK ADDRESS CONTROL (NTW_ADR_CTRL)	32	RW	See description.
2Ch	NETWORK ADDRESS 0 (NTW_ADR_0)	32	RW	5555_5555h

Table continues on the next page...

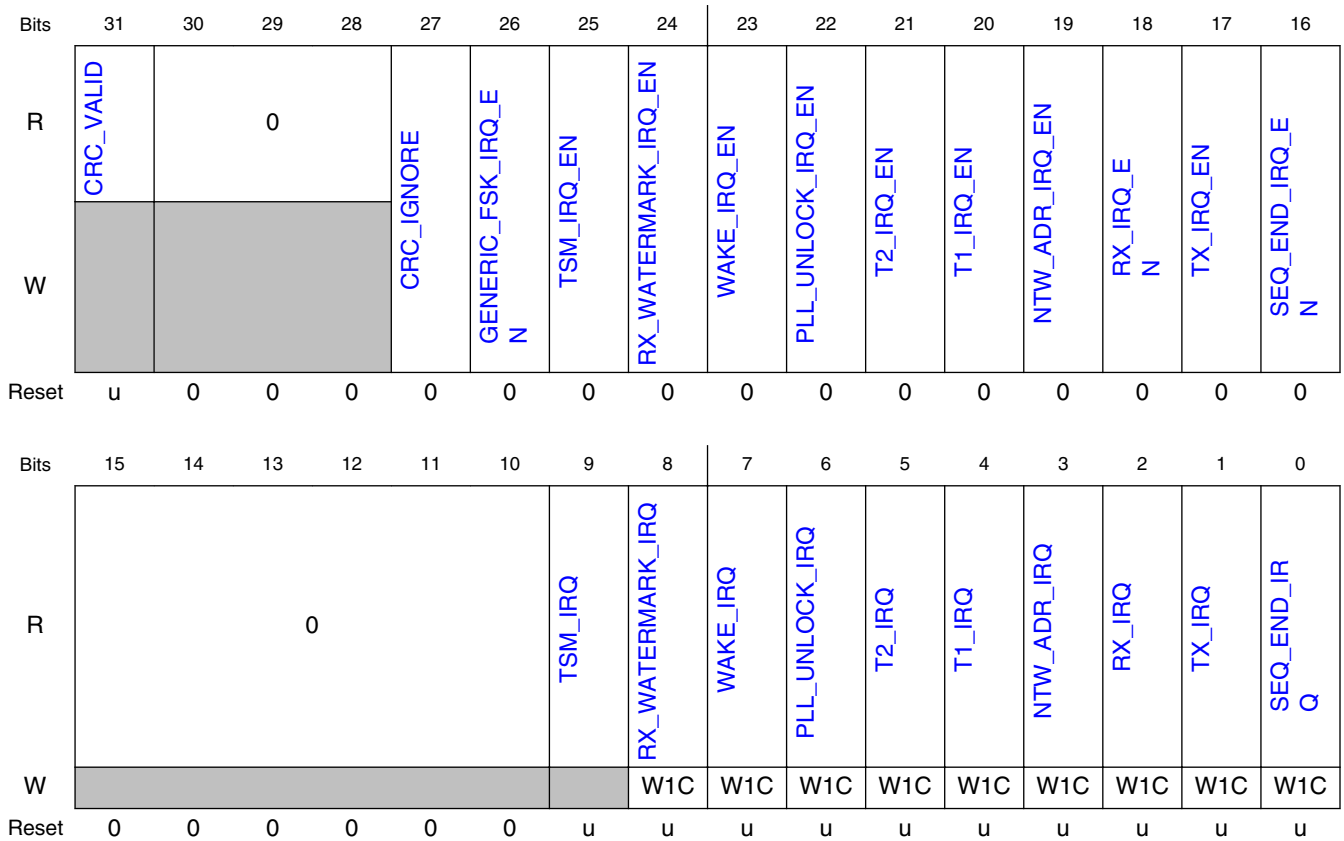
Offset	Register	Width (In bits)	Access	Reset value
30h	NETWORK ADDRESS 1 (NTW_ADR_1)	32	RW	5555_5555h
34h	NETWORK ADDRESS 2 (NTW_ADR_2)	32	RW	5555_5555h
38h	NETWORK ADDRESS 3 (NTW_ADR_3)	32	RW	5555_5555h
3Ch	RECEIVE WATERMARK (RX_WATERMARK)	32	RW	See description.
40h	DSM CONTROL (DSM_CTRL)	32	RW	0000_0000h
44h	PART ID (PART_ID)	32	RO	0000_0001h
60h	PACKET CONFIGURATION (PACKET_CFG)	32	RW	See description.
64h	H0 CONFIGURATION (H0_CFG)	32	RW	0000_0000h
68h	H1 CONFIGURATION (H1_CFG)	32	RW	0000_0000h
6Ch	CRC CONFIGURATION (CRC_CFG)	32	RW	0000_0002h
70h	CRC INITIALIZATION (CRC_INIT)	32	RW	0000_0000h
74h	CRC POLYNOMIAL (CRC_POLY)	32	RW	1021_0000h
78h	CRC XOR OUT (CRC_XOR_OUT)	32	RW	0000_0000h
7Ch	WHITENER CONFIGURATION (WHITEN_CFG)	32	RW	01FF_0918h
80h	WHITENER POLYNOMIAL (WHITEN_POLY)	32	RW	0000_0021h
84h	WHITENER SIZE THRESHOLD (WHITEN_SZ_THR)	32	RW	0000_0800h
88h	BIT RATE (BITRATE)	32	RW	0000_0000h
8Ch	PACKET BUFFER PARTITION POINT (PB_PARTITION)	32	RW	0000_0220h
700h - F7Eh	PACKET BUFFER (PACKET_BUFFER_0 - PACKET_BUFFER_1087)	16	RW	See description.

A.3.3.2 IRQ CONTROL (IRQ_CTRL)

A.3.3.2.1 Offset

Register	Offset
IRQ_CTRL	0h

A.3.3.2.2 Diagram



A.3.3.2.3 Fields

Field	Function
31 CRC_VALID	CRC Valid CRC Valid indicator for RX packets. This bit becomes valid at RX_IRQ, and remains valid until the start of the next RX TSM sequence. 0b - CRC of RX packet is not valid. 1b - CRC of RX packet is valid.
30-28 —	Reserved.
27 CRC_IGNORE	CRC Ignore If set, assert RX_IRQ even for a received packet which fails CRC verification. 0b - RX_IRQ will not be asserted for a received packet which fails CRC verification. 1b - RX_IRQ will be asserted even for a received packet which fails CRC verification.
26 GENERIC_FSK_IRQ_EN	GENERIC_FSK_IRQ Master Enable Master enable for the GENERIC_FSK_IRQ interrupt line to the MCU. 0b - All GENERIC_FSK Interrupts are disabled. 1b - All GENERIC_FSK Interrupts can be enabled.
25	TSM_IRQ Enable

Table continues on the next page...

Field	Function
TSM_IRQ_EN	0b - TSM Interrupt is not enabled. 1b - TSM Interrupt is enabled.
24 RX_WATERMARK_IRQ_EN	RX_WATERMARK_IRQ Enable 0b - RX Watermark Interrupt is not enabled. 1b - RX Watermark Interrupt is enabled.
23 WAKE_IRQ_EN	WAKE_IRQ Enable 0b - Wake Interrupt is not enabled. 1b - Wake Interrupt is enabled.
22 PLL_UNLOCK_IRQ_EN	PLL_UNLOCK_IRQ Enable 0b - PLL Unlock Interrupt is not enabled. 1b - PLL Unlock Interrupt is enabled.
21 T2_IRQ_EN	T2_IRQ Enable 0b - Timer1 (T2) Compare Interrupt is not enabled. 1b - Timer1 (T2) Compare Interrupt is enabled.
20 T1_IRQ_EN	T1_IRQ Enable 0b - Timer1 (T1) Compare Interrupt is not enabled. 1b - Timer1 (T1) Compare Interrupt is enabled.
19 NTW_ADR_IRQ_EN	NTW_ADR_IRQ Enable 0b - Network Address Match Interrupt is not enabled. 1b - Network Address Match Interrupt is enabled.
18 RX_IRQ_EN	RX_IRQ Enable 0b - RX Interrupt is not enabled. 1b - RX Interrupt is enabled.
17 TX_IRQ_EN	TX_IRQ Enable 0b - TX Interrupt is not enabled. 1b - TX Interrupt is enabled.
16 SEQ_END_IRQ_EN	SEQ_END_IRQ Enable 0b - Sequence End Interrupt is not enabled. 1b - Sequence End Interrupt is enabled.
15-10 —	Reserved.
9 TSM_IRQ	TSM Interrupt Indicates TSM0_IRQ or TSM1_IRQ is set in XCVR_STATUS. Clear the bits there. For debug purposes. 0b - TSM0_IRQ and TSM1_IRQ are both clear. 1b - Indicates TSM0_IRQ or TSM1_IRQ is set in XCVR_STATUS.
8 RX_WATERMARK_IRQ	RX Watermark Interrupt Asserts when RX Byte Counter == RX_WATERMARK[12:0] 0b - RX Watermark Interrupt is not asserted. 1b - RX Watermark Interrupt is asserted.
7 WAKE_IRQ	Wake Interrrupt A WAKE_IRQ will be triggered when the GENERIC_FSK Link Layer Controller has awoken from a Manual DSM (Deep Sleep Mode) cycle. WAKE_IRQ indicates that the RF Oscillator has been restarted, and the GENERIC_FSK EVENT_TMR has resumed counting. 0b - Wake Interrupt is not asserted. 1b - Wake Interrupt is asserted.
6	PLL Unlock Interrupt An unlock event has occurred.

Table continues on the next page...

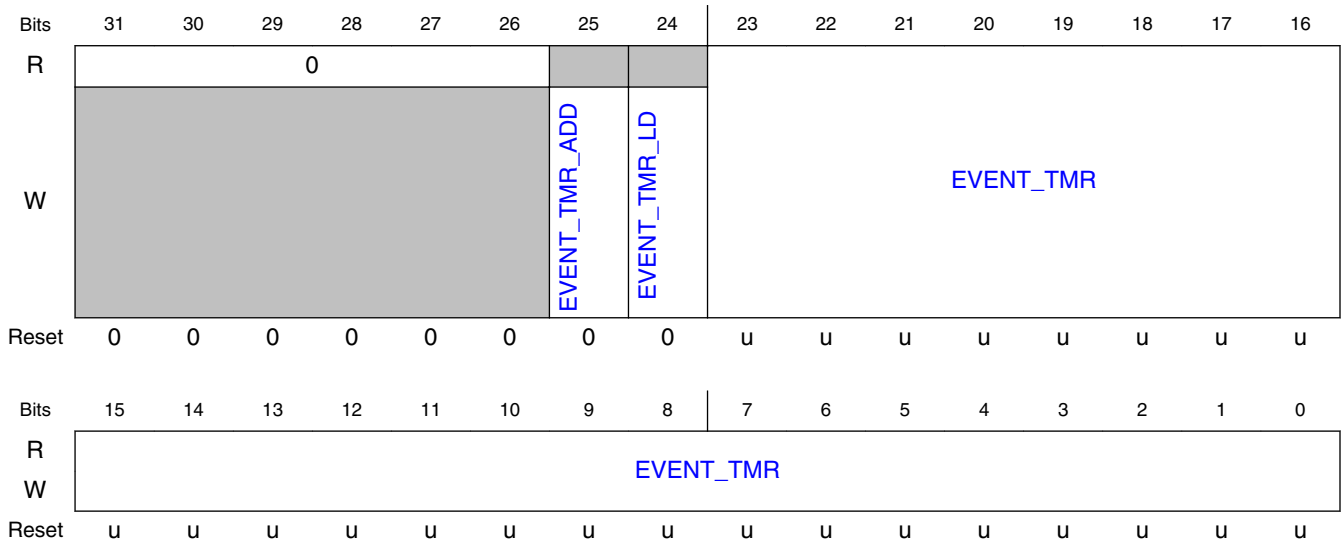
Field	Function
PLL_UNLOCK_IRQ	0b - PLL Unlock Interrupt is not asserted. 1b - PLL Unlock Interrupt is asserted.
5 T2_IRQ	Timer2 (T2) Compare Interrupt 0b - Timer2 (T2) Compare Interrupt is not asserted. 1b - Timer2 (T2) Compare Interrupt is asserted.
4 T1_IRQ	Timer1 (T1) Compare Interrupt 0b - Timer1 (T1) Compare Interrupt is not asserted. 1b - Timer1 (T1) Compare Interrupt is asserted.
3 NTW_ADR_IRQ	Network Address Match Interrupt A Network Address Match has occurred. 0b - Network Address Match Interrupt is not asserted. 1b - Network Address Match Interrupt is asserted.
2 RX_IRQ	RX Interrupt The RX sequence has completed with a successful packet reception. 0b - RX Interrupt is not asserted. 1b - RX Interrupt is asserted.
1 TX_IRQ	TX Interrupt The TX sequence has completed with a successful packet transmission. 0b - TX Interrupt is not asserted. 1b - TX Interrupt is asserted.
0 SEQ_END_IRQ	Sequence End Interrupt Will assert when any TX or RX sequence ends for any reason. 0b - Sequence End Interrupt is not asserted. 1b - Sequence End Interrupt is asserted.

A.3.3.3 EVENT TIMER (EVENT_TMR)

A.3.3.3.1 Offset

Register	Offset
EVENT_TMR	4h

A.3.3.3.2 Diagram



A.3.3.3.3 Fields

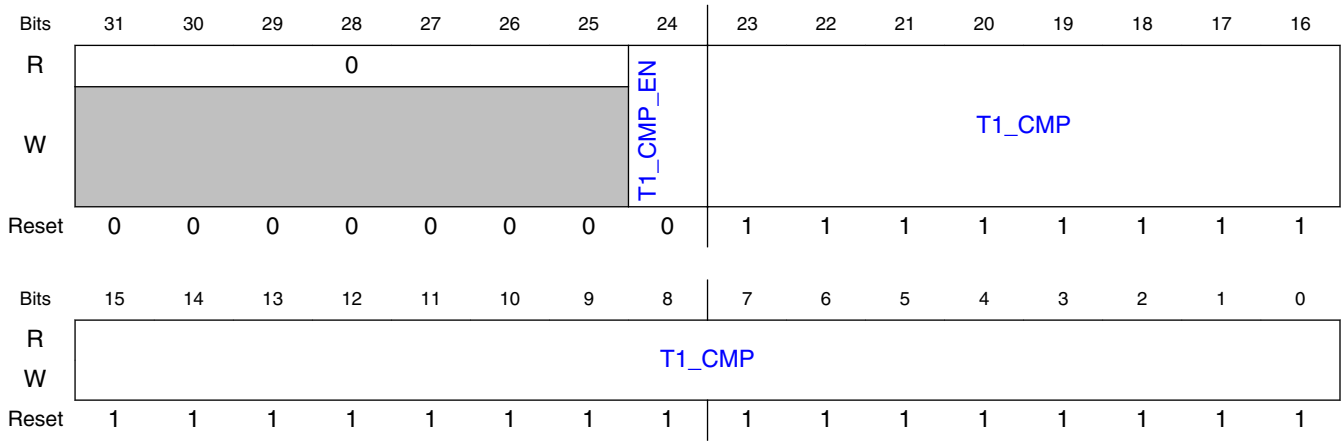
Field	Function
31-26 —	Reserved.
25 EVENT_TMR_ADD	Event Timer Add A write access with this bit increments EVENT_TMR by the contents of EVENT_TMR[23:0]. This is a signed addition.
24 EVENT_TMR_LD	Event Timer Load A write access with this bit set loads EVENT_TMR with the contents of EVENT_TMR[23:0]
23-0 EVENT_TMR	Event Timer Event Timer can be read in these byte locations. To update the Event Timer, either: 1. Write the desired EVENT_TMR to these bytes and set EVENT_TMR_LD=1, or, 2. Write the desired EVENT_TMR increment amount to these bytes and set EVENT_TMR_ADD=1. **Note:** for EVENT_TMR_ADD, EVENT_TMR[23:0] is a signed, two's-complement value.

A.3.3.4 T1 COMPARE (T1_CMP)

A.3.3.4.1 Offset

Register	Offset
T1_CMP	8h

A.3.3.4.2 Diagram



A.3.3.4.3 Fields

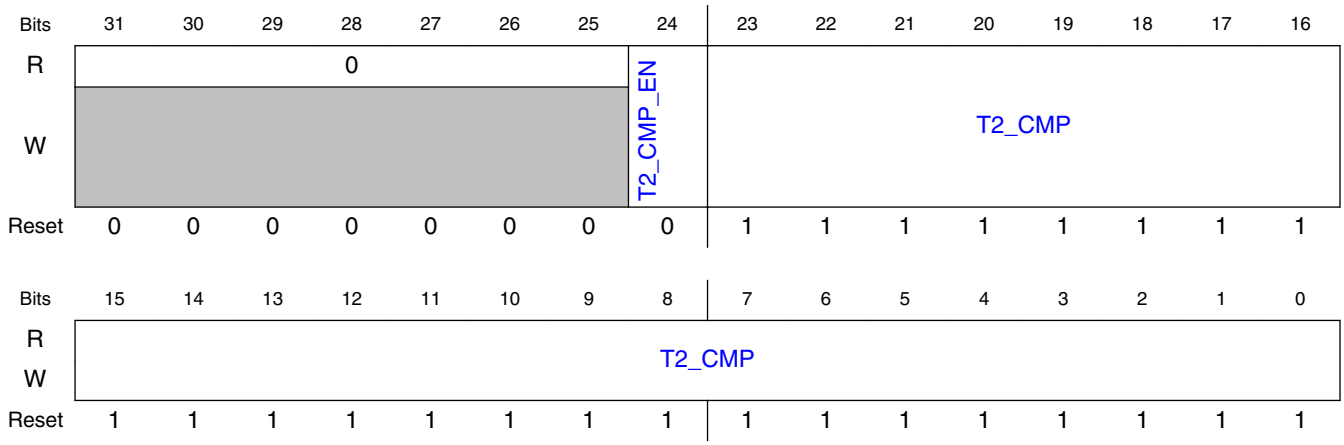
Field	Function
31-25 —	Reserved.
24 T1_CMP_EN	Timer1 (T1) Compare Enable Enable Timer Compare #1 (T1_CMP) to generate T1_IRQ and/or execute Sequence Commands.
23-0 T1_CMP	Timer1 (T1) Compare Value Timer1 (T1) Compare Value. Can be used to generate T1_IRQ and/or launch Sequence Commands

A.3.3.5 T2 COMPARE (T2_CMP)

A.3.3.5.1 Offset

Register	Offset
T2_CMP	Ch

A.3.3.5.2 Diagram



A.3.3.5.3 Fields

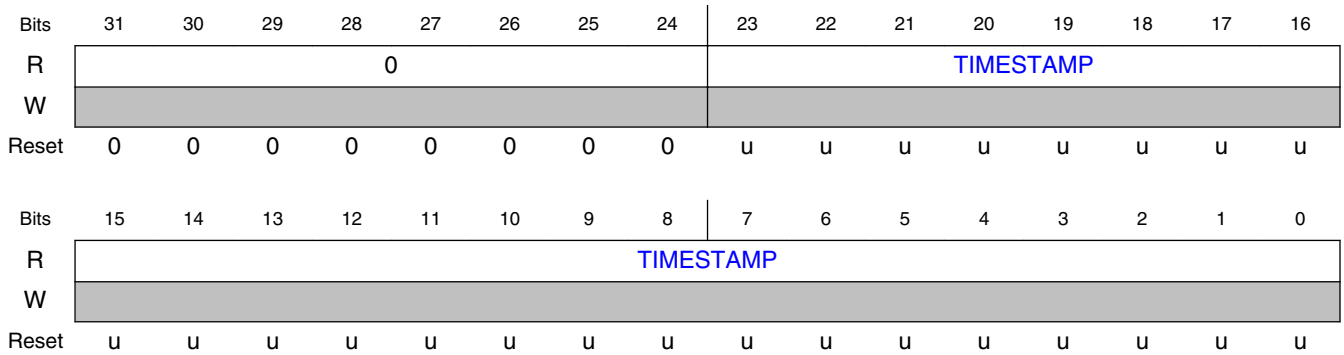
Field	Function
31-25 —	Reserved.
24 T2_CMP_EN	Timer2 (T2) Compare Enable Enable Timer Compare #2 (T2_CMP) to generate T2_IRQ and/or execute Sequence Commands.
23-0 T2_CMP	Timer2 (T2) Compare Value Timer2 (T2) Compare Value. Can be used to generate T2_IRQ and/or launch Sequence Commands

A.3.3.6 TIMESTAMP (TIMESTAMP)

A.3.3.6.1 Offset

Register	Offset
TIMESTAMP	10h

A.3.3.6.2 Diagram



A.3.3.6.3 Fields

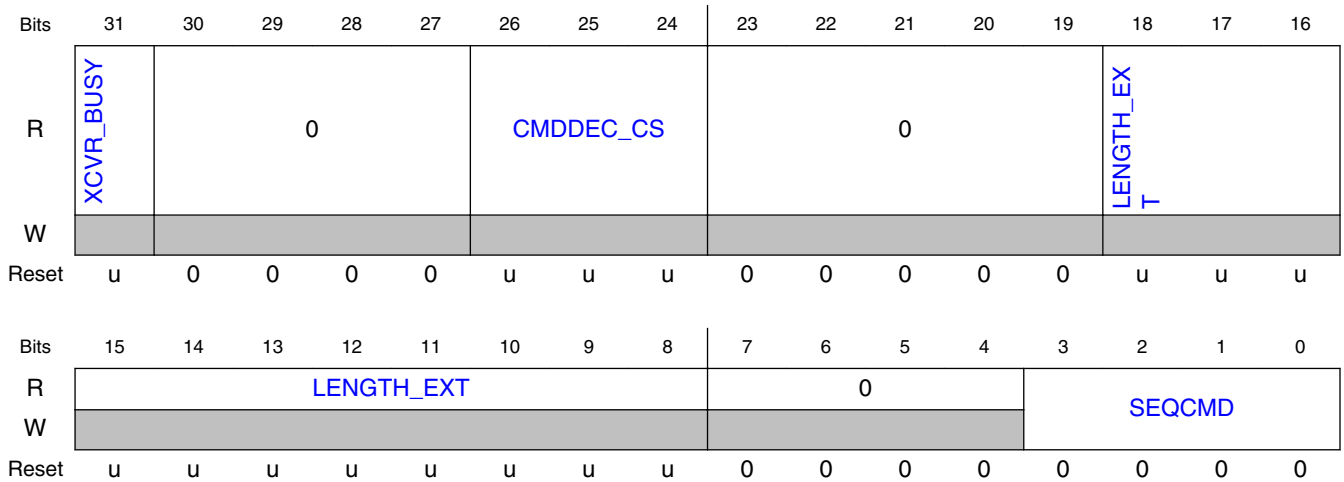
Field	Function
31-24 —	Reserved.
23-0 TIMESTAMP	Received Packet Timestamp Received Packet Timestamp. Captured at NTW_ADR_IRQ.

A.3.3.7 TRANSCEIVER CONTROL (XCVR_CTRL)

A.3.3.7.1 Offset

Register	Offset
XCVR_CTRL	14h

A.3.3.7.2 Diagram



A.3.3.7.3 Fields

Field	Function
31 XCVR_BUSY	Transceiver Busy For multi-protocol arbitration, XCVR_BUSY=1 indicates an RX or TX operation is underway, by either GENERIC_FSK, or some other protocol. 0b - IDLE 1b - BUSY
30-27 —	Reserved.
26-24 CMDDEC_CS	Command Decode Current State of the Command Decoder FSM (debug only)
23-19 —	Reserved.
18-8 LENGTH_EXT	Extracted Length Field The read-only register reflects the contents of the LENGTH field of the Header of the most-recently received packet. This is the raw, unmodified LENGTH field, as it appears in the received bit stream, unmodified by LENGTH_ADJ or any other parameter.
7-4 —	Reserved.
3-0 SEQCMD	Sequence Commands 0000b - No Action 0001b - TX Start Now 0010b - TX Start @ T1 Timer Compare Match (EVENT_TMR = T1_CMP) 0011b - TX Start @ T2 Timer Compare Match (EVENT_TMR = T2_CMP) 0100b - TX Cancel -- Cancels pending TX events but do not abort a TX-in-progress 0101b - RX Start Now 0110b - RX Start @ T1 Timer Compare Match (EVENT_TMR = T1_CMP) 0111b - RX Start @ T2 Timer Compare Match (EVENT_TMR = T2_CMP) 1000b - RX Stop @ T1 Timer Compare Match (EVENT_TMR = T1_CMP)

Field	Function
	1001b - RX Stop @ T2 Timer Compare Match (EVENT_TMR = T2_CMP) 1010b - RX Cancel -- Cancels pending RX events but do not abort a RX-in-progress 1011b - Abort All - Cancels all pending events and abort any sequence-in-progress 1100b - Reserved 1101b - Reserved 1110b - Reserved 1111b - Reserved

A.3.3.8 TRANSCEIVER STATUS (XCVR_STS)

A.3.3.8.1 Offset

Register	Offset
XCVR_STS	18h

A.3.3.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LQI								RSSI							
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CRC_VALID	LQI_VALID	0	RX_IN_WARMDN	RX_IN_PROGRESS	RX_IN_SEARCH	RX_IN_WARMUP	RX_STOP_T2_PEND	RX_STOP_T1_PEND	RX_START_T2_PEND	RX_START_T1_PEND	TX_IN_WARMDN	TX_IN_PROGRESS	TX_IN_WARMUP	TX_START_T2_PEND	TX_START_T1_PEND
W																
Reset	u	u	0	u	u	u	u	u	u	u	u	u	u	u	u	u

A.3.3.8.3 Fields

Field	Function
31-24 LQI	Link Quality Indicator This field is valid when LQI_VALID=1. LQI is a unsigned, unitless value.

Table continues on the next page...

Field	Function
23-16 RSSI	Received Signal Stength Indicator, in dBm
15 CRC_VALID	CRC Valid Indicator CRC Valid indicator for RX packets. 0b - CRC is not valid for RX packet. 1b - CRC is valid for RX packet.
14 LQI_VALID	LQI Valid Indicator LQI Valid indicator for RX packets. This bit becomes set when the LQI computation completes for the packet currently being received, and remains set for the remainder of the packet. 0b - LQI is not yet valid for RX packet. 1b - LQI is valid for RX packet.
13 —	Reserved.
12 RX_IN_WARMDN	RX Warmdown Status RX Sequence in TSM Warmdown
11 RX_IN_PROGRESS	RX in Progress Status RX Packet Reception Currently Underway
10 RX_IN_SEARCH	RX Search Status RX Sequence in Network Address Search
9 RX_IN_WARMUP	RX Warmup Status RX Sequence in TSM Warmup
8 RX_STOP_T2_PEND	RX T2 Start Pending Status RX Sequence will stop @ next T2 Match
7 RX_STOP_T1_PEND	RX T1 Stop Pending Status RX Sequence will stop @ next T1 Match
6 RX_START_T2_PEND	RX T2 Start Pending Status RX Sequence will start @ next T2 Match
5 RX_START_T1_PEND	RX T1 Start Pending Status RX Sequence will start @ next T1 Match
4 TX_IN_WARMDN	TX Warmdown Status TX Sequence in TSM Warmdown
3 TX_IN_PROGRESS	TX in Progress Status TX Packet Transmission Currently Underway
2	TX Warmup Status

Table continues on the next page...

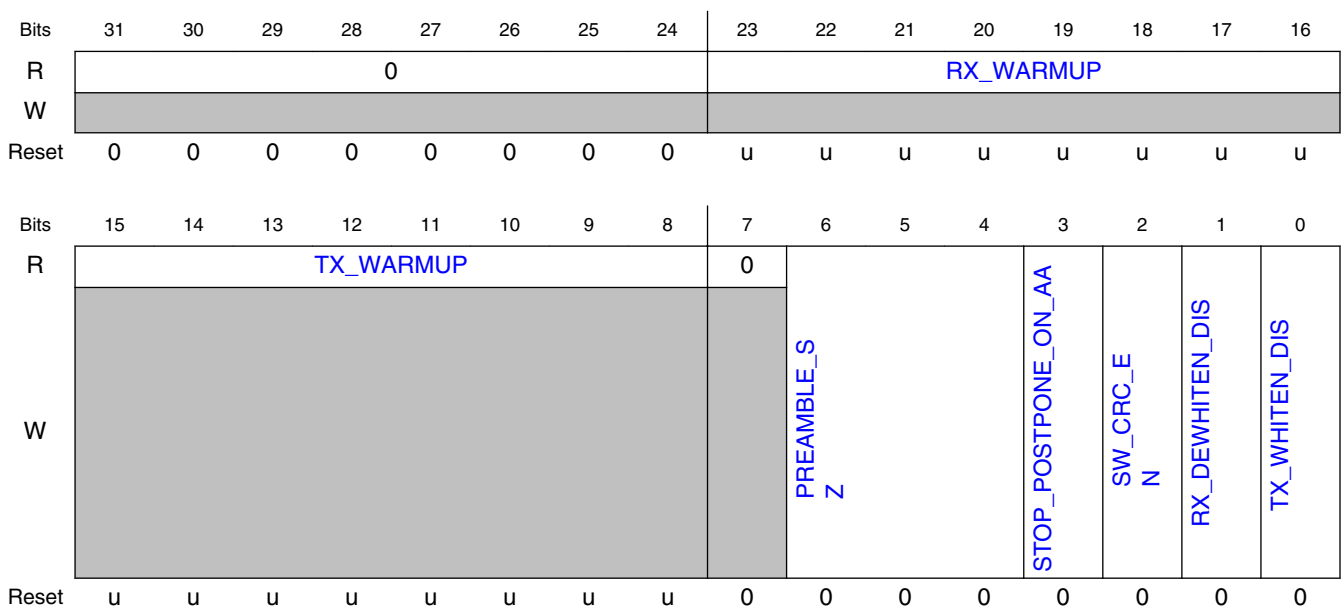
Field	Function
TX_IN_WARMUP	TX Sequence in TSM Warmup
1 TX_START_T2_PEND	TX T2 Start Pending Status TX Sequence will start @ next T2 Match
0 TX_START_T1_PEND	TX T1 Start Pending Status TX Sequence will start @ next T1 Match

A.3.3.9 TRANSCEIVER CONFIGURATION (XCVR_CFG)

A.3.3.9.1 Offset

Register	Offset
XCVR_CFG	1Ch

A.3.3.9.2 Diagram



A.3.3.9.3 Fields

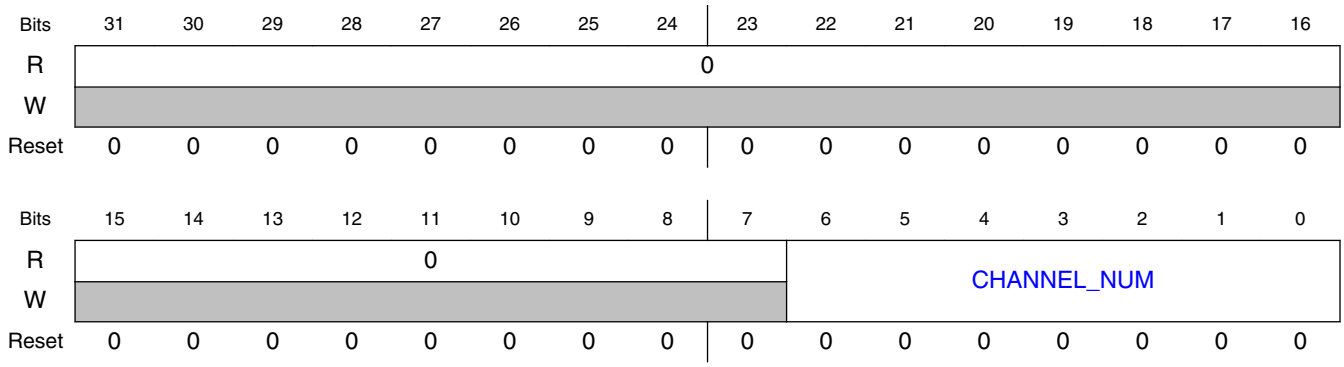
Field	Function
31-24 —	Reserved.
23-16 RX_WARMUP	Receive Warmup Time RX warmup time, in microseconds, provided to Link Layer software, so that warmup time can be accounted for when scheduling over-the-air transactions.
15-8 TX_WARMUP	Transmit Warmup Time TX warmup time, in microseconds, provided to Link Layer software, so that warmup time can be accounted for when scheduling over-the-air transactions.
7 —	Reserved.
6-4 PREAMBLE_SZ	Preamble Size Number of Octets = PREAMBLE_SZ + 1, where 0 <= PREAMBLE_SZ <= 7
3 STOP_POSTPONE_ON_AA	Postpone Stop Command Timeout On Access Address Match Enable If this bit is set, if a RX STOP condition occurs, due to an Event Timer match to a previously issued RX_STOP_T1 or RX_STOP_T2 sequence command, while a packet is being received (*ntw_adr_matched* = 1), the timeout-related abort will be deferred untl the packet reception is completed. If the packet is good, HW will signal Data Indication; otherwise the Sequence Manager will return to IDLE and await further commanding. 0b - STOP Abort will occur on RX_STOP_T1 or RX_STOP_T1 Event Timer match, regardless of ntw_adr_matched 1b - STOP Abort will be deferred on RX_STOP_T1 or RX_STOP_T1 Event Timer match, if ntw_adr_matched is asserted; otherwise the RX_STOP Abort will occur immediately
2 SW_CRC_EN	Software CRC Enable Software override of the HW-computed CRC for TX. Software must write CRC to Packet Buffer (RAM)
1 RX_DEWHITEN_DIS	RX De-Whitening Disable Disable all de-whitening on RX packets
0 TX_WHITEN_DIS	TX Whitening Disable Disable all whitening on TX packets

A.3.3.10 CHANNEL NUMBER (CHANNEL_NUM)

A.3.3.10.1 Offset

Register	Offset
CHANNEL_NUM	20h

A.3.3.10.2 Diagram



A.3.3.10.3 Fields

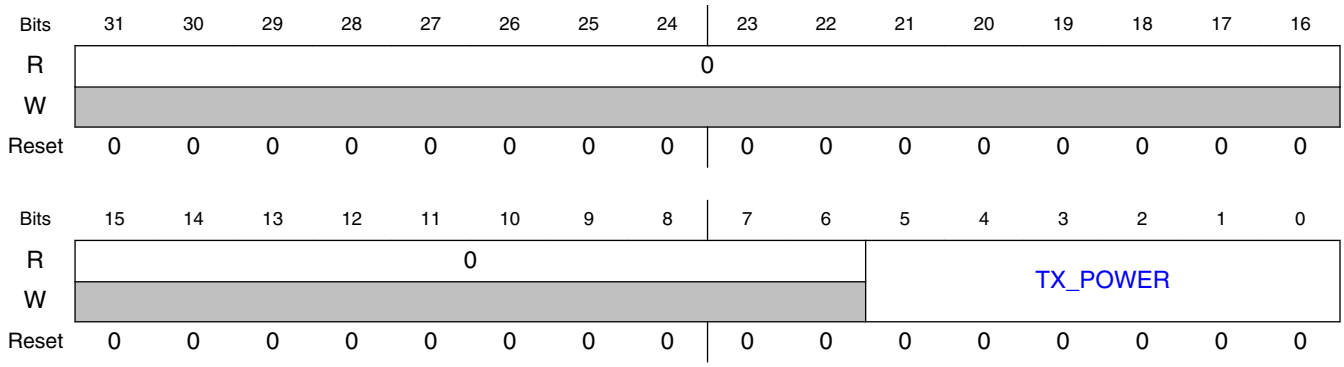
Field	Function
31-7 —	Reserved.
6-0 CHANNEL_NUM	Channel Number RF Channel Select: 0 <= CHANNEL_NUM <= 127; Formula: F = (2360 + CHANNEL_NUM) [in MHz]

A.3.3.11 TRANSMIT POWER (TX_POWER)

A.3.3.11.1 Offset

Register	Offset
TX_POWER	24h

A.3.3.11.2 Diagram



A.3.3.11.3 Fields

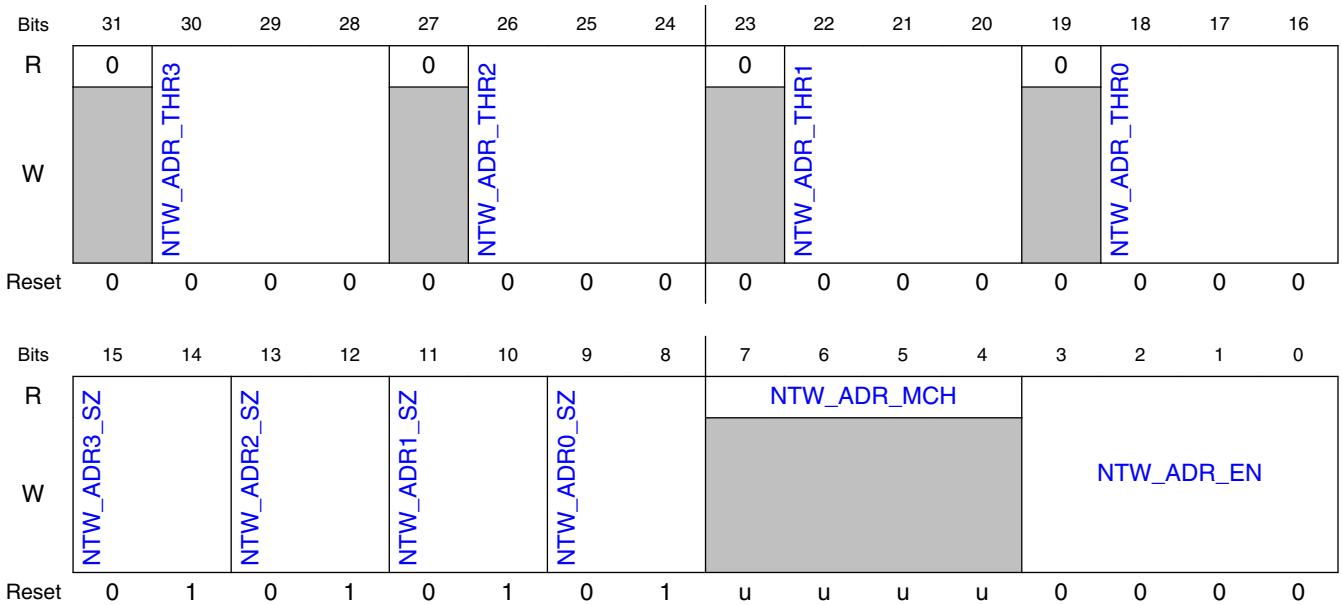
Field	Function
31-6 —	Reserved.
5-0 TX_POWER	Transmit Power PA Power Level.

A.3.3.12 NETWORK ADDRESS CONTROL (NTW_ADR_CTRL)

A.3.3.12.1 Offset

Register	Offset
NTW_ADR_CTRL	28h

A.3.3.12.2 Diagram



A.3.3.12.3 Fields

Field	Function
31 —	Reserved.
30-28 NTW_ADR_THR3	Network Address 3 Threshold Number of Tolerated bit errors for Network Address 3
27 —	Reserved.
26-24 NTW_ADR_THR2	Network Address 2 Threshold Number of Tolerated bit errors for Network Address 2
23 —	Reserved.
22-20 NTW_ADR_THR1	Network Address 1 Threshold Number of Tolerated bit errors for Network Address 1
19 —	Reserved.
18-16 NTW_ADR_THR0	Network Address 0 Threshold Number of Tolerated bit errors for Network Address 0
15-14	Network Address 3 Size

Table continues on the next page...

Field	Function
NTW_ADR3_SZ	00b - Network Address 3 requires a 8-bit correlation 01b - Network Address 3 requires a 16-bit correlation 10b - Network Address 3 requires a 24-bit correlation 11b - Network Address 3 requires a 32-bit correlation
13-12 NTW_ADR2_SZ	Network Address 2 Size 00b - Network Address 2 requires a 8-bit correlation 01b - Network Address 2 requires a 16-bit correlation 10b - Network Address 2 requires a 24-bit correlation 11b - Network Address 2 requires a 32-bit correlation
11-10 NTW_ADR1_SZ	Network Address 1 Size 00b - Network Address 1 requires a 8-bit correlation 01b - Network Address 1 requires a 16-bit correlation 10b - Network Address 1 requires a 24-bit correlation 11b - Network Address 1 requires a 32-bit correlation
9-8 NTW_ADR0_SZ	Network Address 0 Size 00b - Network Address 0 requires a 8-bit correlation 01b - Network Address 0 requires a 16-bit correlation 10b - Network Address 0 requires a 24-bit correlation 11b - Network Address 0 requires a 32-bit correlation
7-4 NTW_ADR_MCH	Network Address Match Indicates which of the 4 Network Addresses has matched in the PHY. Valid during an RX sequence at the point of match, and remains asserted until either: 1. The next RX sequence begins (if the current packet passed CRC and header filtering), or, 2. An RX recycle to Network Address search (if the current packet failed CRC or header filtering) 0001b - Network Address 0 has matched 0010b - Network Address 1 has matched 0100b - Network Address 2 has matched 1000b - Network Address 3 has matched
3-0 NTW_ADR_EN	Network Address Enable Enable Network Address N for PHY correlation, where 0 ≤ N ≤ 3. Any bit combination can be set. 0001b - Enable Network Address 0 for correlation 0010b - Enable Network Address 1 for correlation 0100b - Enable Network Address 2 for correlation 1000b - Enable Network Address 3 for correlation

A.3.3.13 NETWORK ADDRESS 0 (NTW_ADR_0)

A.3.3.13.1 Offset

Register	Offset
NTW_ADR_0	2Ch

A.3.3.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NTW_ADR_0															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NTW_ADR_0															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

A.3.3.13.3 Fields

Field	Function
31-0 NTW_ADR_0	Network Address 0 The PHY will search for this Network Address if NTW_ADR_CTRL[NTW_ADR_EN[0]] = 1

A.3.3.14 NETWORK ADDRESS 1 (NTW_ADR_1)

A.3.3.14.1 Offset

Register	Offset
NTW_ADR_1	30h

A.3.3.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NTW_ADR_1															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NTW_ADR_1															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

A.3.3.14.3 Fields

Field	Function
31-0 NTW_ADR_1	Network Address 1 The PHY will search for this Network Address if NTW_ADR_CTRL[NTW_ADR_EN[1]] = 1

A.3.3.15 NETWORK ADDRESS 2 (NTW_ADR_2)

A.3.3.15.1 Offset

Register	Offset
NTW_ADR_2	34h

A.3.3.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NTW_ADR_2															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NTW_ADR_2															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

A.3.3.15.3 Fields

Field	Function
31-0 NTW_ADR_2	Network Address 2 The PHY will search for this Network Address if NTW_ADR_CTRL[NTW_ADR_EN[2]] = 1

A.3.3.16 NETWORK ADDRESS 3 (NTW_ADR_3)

A.3.3.16.1 Offset

Register	Offset
NTW_ADR_3	38h

A.3.3.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NTW_ADR_3															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NTW_ADR_3															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

A.3.3.16.3 Fields

Field	Function
31-0 NTW_ADR_3	Network Address 2 The PHY will search for this Network Address if NTW_ADR_CTRL[NTW_ADR_EN[3]] = 1

A.3.3.17 RECEIVE WATERMARK (RX_WATERMARK)

A.3.3.17.1 Offset

Register	Offset
RX_WATERMARK	3Ch

A.3.3.17.2 Diagram

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

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

A.3.3.17.3 Fields

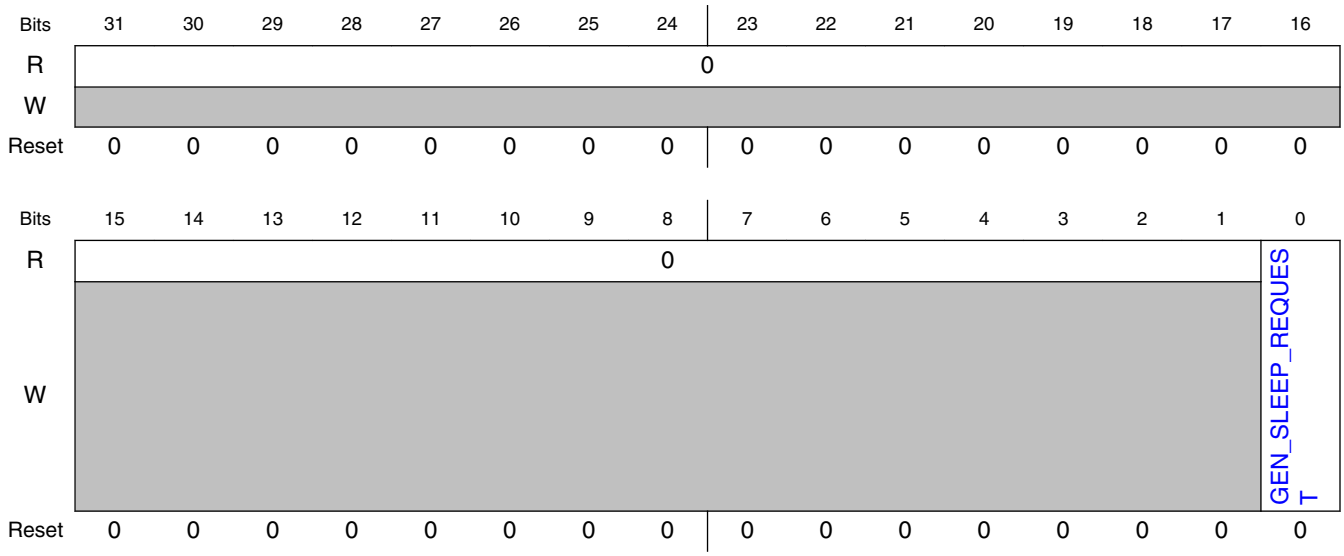
Field	Function
31-29 —	Reserved.
28-16 BYTE_COUNTER	Byte Counter Reflects the current Byte Count, for TX and RX. This is a signed, twos-complement value. For values less than zero, indicates the preamble transmission is underway (TX only). A value of 0 indicates the first octet of Network Address is being transmitted or received. A value of 1 indicates the second octet of Network Address is being transmitted or received. Etc.
15-13 —	Reserved.
12-0 RX_WATERMARK	Receive Watermark Sets the trigger for RX_WATERMARK_IRQ. Trigger the RX_WATERMARK_IRQ when: RX Byte Counter == RX_WATERMARK[12:0]

A.3.3.18 DSM CONTROL (DSM_CTRL)

A.3.3.18.1 Offset

Register	Offset
DSM_CTRL	40h

A.3.3.18.2 Diagram



A.3.3.18.3 Fields

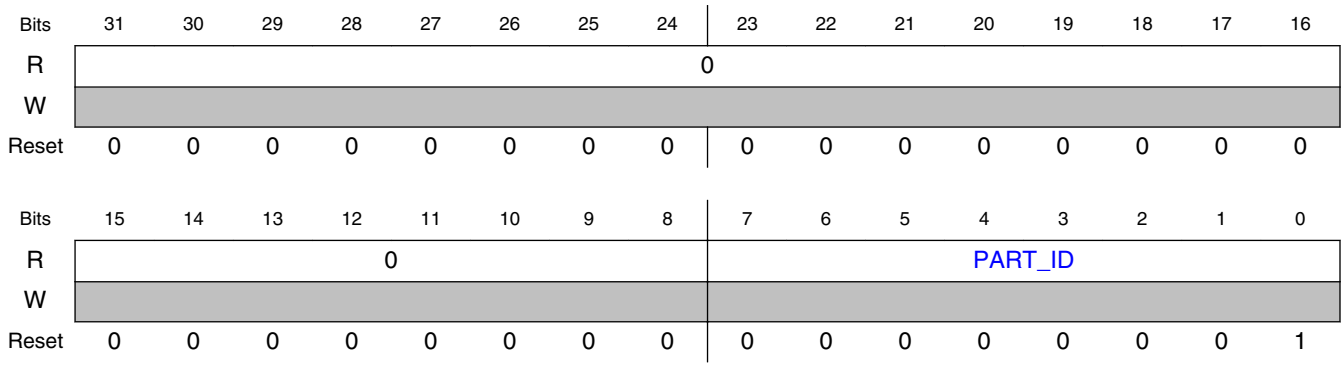
Field	Function
31-1 —	Reserved.
0 GEN_SLEEP_REQUEST	GENERIC_FSK Deep Sleep Mode Request Setting GEN_SLEEP_REQUEST to 1 enables Deep Sleep Mode (DSM) to be entered when the RSIM DSM_TIMER matches the RSIM MAN_SLEEP register.

A.3.3.19 PART ID (PART_ID)

A.3.3.19.1 Offset

Register	Offset
PART_ID	44h

A.3.3.19.2 Diagram



A.3.3.19.3 Fields

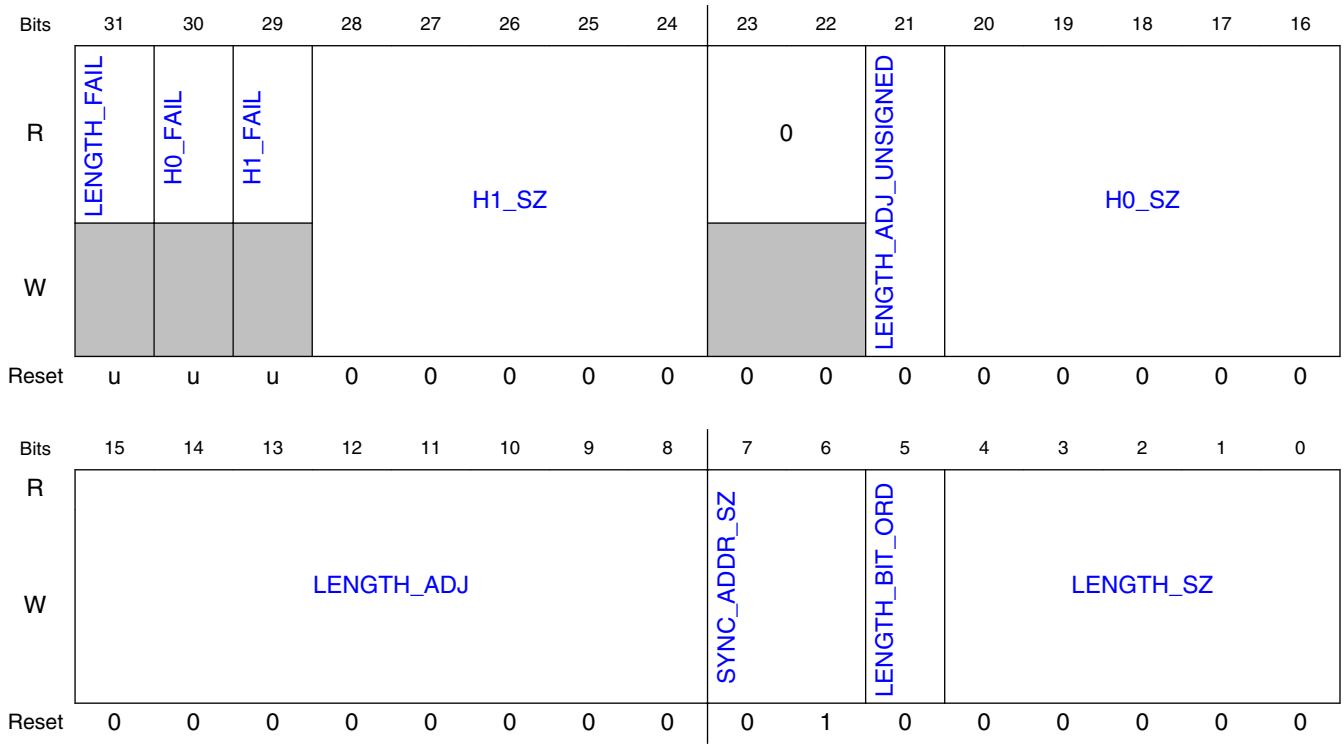
Field	Function
31-8 —	Reserved.
7-0 PART_ID	Part ID Part ID to identify HW revision of the Generic FSK Link Layer

A.3.3.20 PACKET CONFIGURATION (PACKET_CFG)

A.3.3.20.1 Offset

Register	Offset
PACKET_CFG	60h

A.3.3.20.2 Diagram



A.3.3.20.3 Fields

Field	Function
31 LENGTH_FAIL	Maximum Length Violated Status Bit For packets received with REC_BAD_PKT=1, LENGTH_FAIL indicates the extracted LENGTH header field exceeded LENGTH_MAX
30 H0_FAIL	H0 Violated Status Bit For packets received with REC_BAD_PKT=1, H0_FAIL indicates the received H0 header field violates the H0_MASK/H0_MATCH pattern
29 H1_FAIL	H1 Violated Status Bit For packets received with REC_BAD_PKT=1, H1_FAIL indicates the received H1 header field violates the H1_MASK/H1_MATCH pattern
28-24 H1_SZ	H1 Size Length of H1 in bits; 0 ≤ H1_SZ ≤ 16
23-22 —	Reserved.
21 LENGTH_ADJ_UNSIGNED	Length Adjustment Unsigned Enabled 0b - Hardware interprets LENGTH_ADJ as a signed integer (default) 1b - Hardware interprets LENGTH_ADJ as a unsigned integer
20-16 H0_SZ	H0 Size Size of H0 in bits; 0 ≤ H0_SZ ≤ 16

Table continues on the next page...

Field	Function
15-8 LENGTH_ADJ	Length Adjustment Signed Adjustment to the LENGTH field for TX and RX. A value of 0 (default) means LENGTH is interpreted as PAYLOAD + CRC
7-6 SYNC_ADDR_SZ	Sync Address Size Number of Octets = SYNC_ADDR_SZ + 1, 0 ≤ SYNC_ADDR_SZ ≤ 3.
5 LENGTH_BIT_ORD	LENGTH Bit Order Bit order for the LENGTH field of the header 0b - LS Bit First 1b - MS Bit First
4-0 LENGTH_SZ	LENGTH Size Size of LENGTH field of the header, in bits; 0 ≤ LENGTH_SZ ≤ 16

A.3.3.21 H0 CONFIGURATION (H0_CFG)

A.3.3.21.1 Offset

Register	Offset
H0_CFG	64h

A.3.3.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	H0_MASK															
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	H0_MATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.3.3.21.3 Fields

Field	Function
31-16 H0_MASK	H0 Mask Register

Table continues on the next page...

Field	Function
	For each bit that is set to 1, the received H0 field must match the corresponding bit of H0_MATCH[15:0], else the received packet is rejected.
15-0 H0_MATCH	H0 Match Register For each bit of H0_MASK[15:0] that is set to 1, the received H0 field must match this register, else the received packet is rejected.

A.3.3.22 H1 CONFIGURATION (H1_CFG)

A.3.3.22.1 Offset

Register	Offset
H1_CFG	68h

A.3.3.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	H1_MASK															
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	H1_MATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A.3.3.22.3 Fields

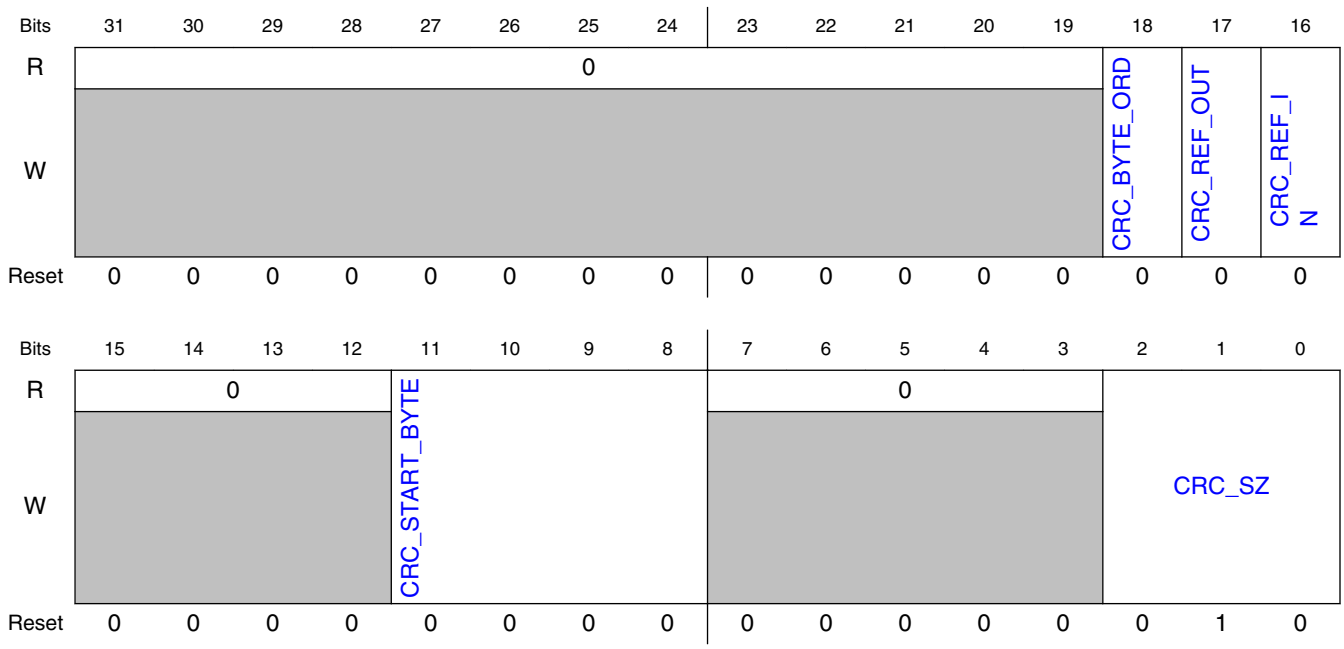
Field	Function
31-16 H1_MASK	H1 Mask Register For each bit that is set to 1, the received H1 field must match the corresponding bit of H1_MATCH[15:0], else the received packet is rejected.
15-0 H1_MATCH	H1 Match Register For each bit of H1_MASK[15:0] that is set to 1, the received H1 field must match this register, else the received packet is rejected.

A.3.3.23 CRC CONFIGURATION (CRC_CFG)

A.3.3.23.1 Offset

Register	Offset
CRC_CFG	6Ch

A.3.3.23.2 Diagram



A.3.3.23.3 Fields

Field	Function
31-19 —	Reserved.
18 CRC_BYTE_ORD	CRC Byte Order 0b - LS Byte First 1b - MS Byte First
17 CRC_REF_OUT	CRC Reflect Out 0b - do not manipulate CRC result 1b - CRC result is to be reflected bitwise (operated on entire word)
16 CRC_REF_IN	CRC Reflect In 0b - do not manipulate input data stream 1b - reflect each byte in the input stream bitwise

Table continues on the next page...

Field	Function
15-12 —	Reserved.
11-8 CRC_START_BYTE	Configure CRC Start Point Start CRC with this byte position. Byte #0 is the first byte of Sync Address
7-3 —	Reserved.
2-0 CRC_SZ	CRC Size (in octets) Number of CRC Octets = CRC_SZ, 0 ≤ CRC_SZ ≤ 4.

A.3.3.24 CRC INITIALIZATION (CRC_INIT)

A.3.3.24.1 Offset

Register	Offset
CRC_INIT	70h

A.3.3.24.2 Diagram

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

A.3.3.24.3 Fields

Field	Function
31-0 CRC_SEED	CRC Seed Value Initial Value for CRC LFSR

A.3.3.25 CRC POLYNOMIAL (CRC_POLY)

A.3.3.25.1 Offset

Register	Offset
CRC_POLY	74h

A.3.3.25.2 Diagram

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

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

A.3.3.25.3 Fields

Field	Function
31-0 CRC_POLY	CRC Polynomial.

A.3.3.26 CRC XOR OUT (CRC_XOR_OUT)

A.3.3.26.1 Offset

Register	Offset
CRC_XOR_OUT	78h

A.3.3.26.2 Diagram

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

A.3.3.26.3 Fields

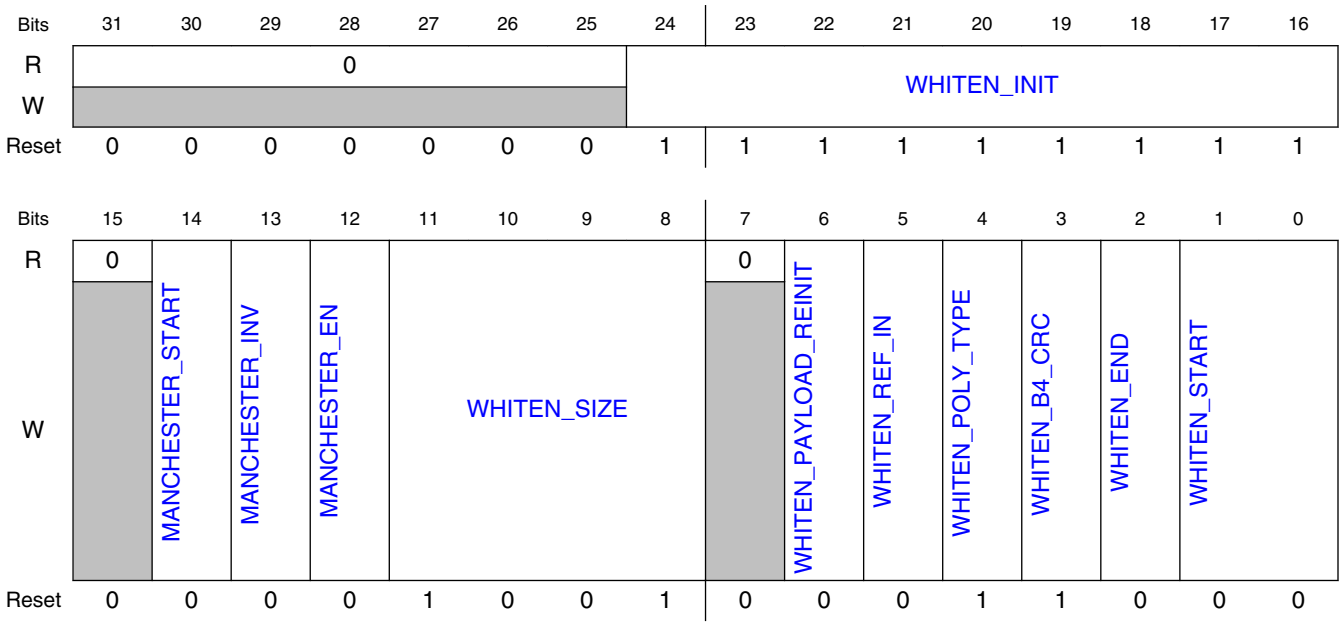
Field	Function
31-0 CRC_XOR_OUT	CRC XOR OUT Register XOR mask for CRC result (for no mask, should be 0)

A.3.3.27 WHITENER CONFIGURATION (WHITEN_CFG)

A.3.3.27.1 Offset

Register	Offset
WHITEN_CFG	7Ch

A.3.3.27.2 Diagram



A.3.3.27.3 Fields

Field	Function
31-25 —	Reserved.
24-16 WHITEN_INIT	Initialization Value for Whitening/De-whitening
15 —	Reserved.
14 MANCHESTER_START	Configure Manchester Encoding Start Point 0b - Start Manchester coding at start-of-payload 1b - Start Manchester coding at start-of-header
13 MANCHESTER_INV	Configure for Inverted Manchester Encoding 0b - Manchester coding as per 802.3 1b - Manchester coding as per 802.3 but with the encoding signal inverted
12 MANCHESTER_EN	Configure for Manchester Encoding/Decoding 0b - Disable Manchester encoding (TX) and decoding (RX) 1b - Enable Manchester encoding (TX) and decoding (RX)
11-8 WHITEN_SIZE	Length of Whitener LFSR
7 —	Reserved.
6	Configure for Whitener re-initialization

Table continues on the next page...

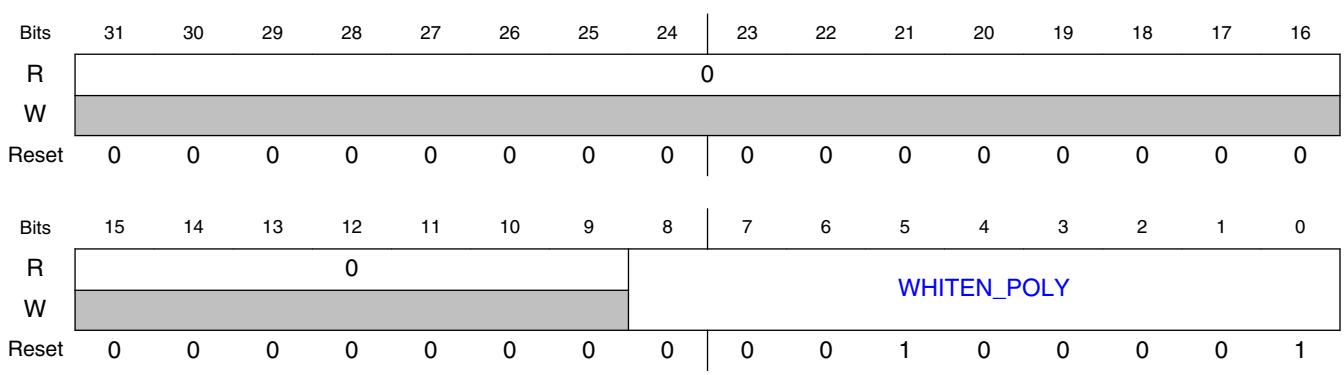
Field	Function
WHITEN_PAYLOAD_REINIT	0b - Don't re-initialize Whitener LFSR at start-of-payload 1b - Re-initialize Whitener LFSR at start-of-payload
5 WHITEN_REF_IN	Whiten Reflect Input The input data stream is reflected, bit-wise, per byte, if this register bit is asserted. Bit 7 becomes bit 0, bit 6 becomes bit 1, etc. This register bit will only cause the reflection of the payload data bits as they are used in the whiten calculation and will not cause any change in the output bit order.
4 WHITEN_POLY_TYPE	Whiten Polynomial Type A Fibonacci type LFSR is used with the whiten polynomial if this register bit is asserted. Otherwise, a Galois type LFSR is used.
3 WHITEN_B4_CRC	Congifure for Whitening-before-CRC Sets the order of Bit Stream Processing for TX and RX. 0b - CRC before whiten/de-whiten 1b - Whiten/de-whiten before CRC
2 WHITEN_END	Configure end-of-whitening 0b - end whiten at end-of-payload 1b - end whiten at end-of-crc
1-0 WHITEN_START	Configure Whitener Start Point 00b - no whitening 01b - start whitening at start-of-H0 10b - start whitening at start-of-H1 but only if LENGTH > WHITEN_SZ_THR 11b - start whitening at start-of-payload but only if LENGTH > WHITEN_SZ_THR

A.3.3.28 WHITENER POLYNOMIAL (WHITEN_POLY)

A.3.3.28.1 Offset

Register	Offset
WHITEN_POLY	80h

A.3.3.28.2 Diagram



A.3.3.28.3 Fields

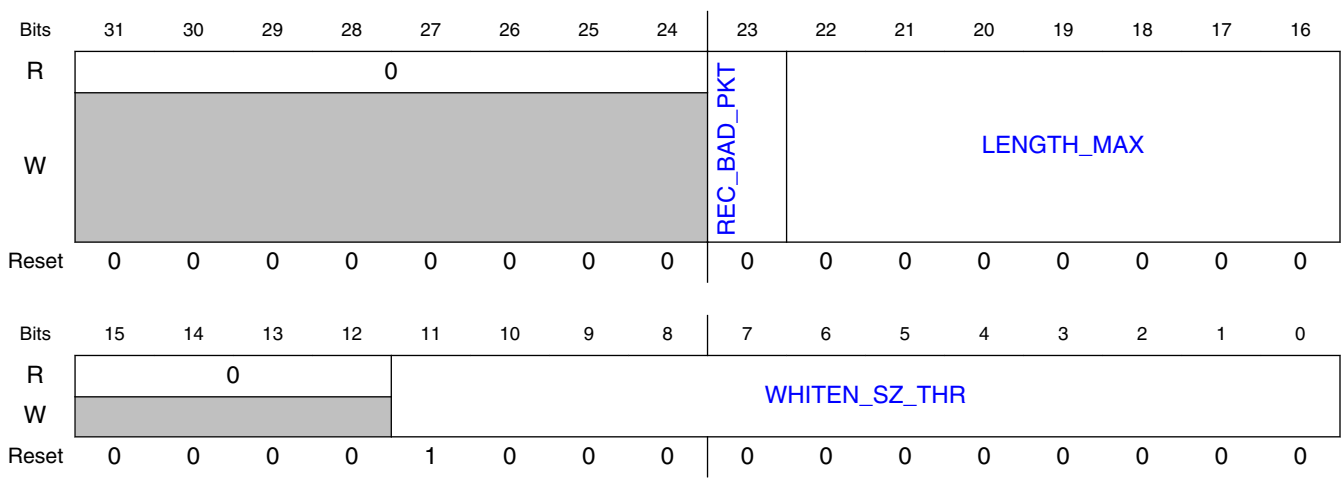
Field	Function
31-9 —	Reserved.
8-0 WHITEN_POLY	Whitener Polynomial The 9-bit polynomial used in the whiten calculation. The polynomial value must be right-justified if smaller than 9-bits.

A.3.3.29 WHITENER SIZE THRESHOLD (WHITEN_SZ_THR)

A.3.3.29.1 Offset

Register	Offset
WHITEN_SZ_THR	84h

A.3.3.29.2 Diagram



A.3.3.29.3 Fields

Field	Function
31-24 —	Reserved.

Table continues on the next page...

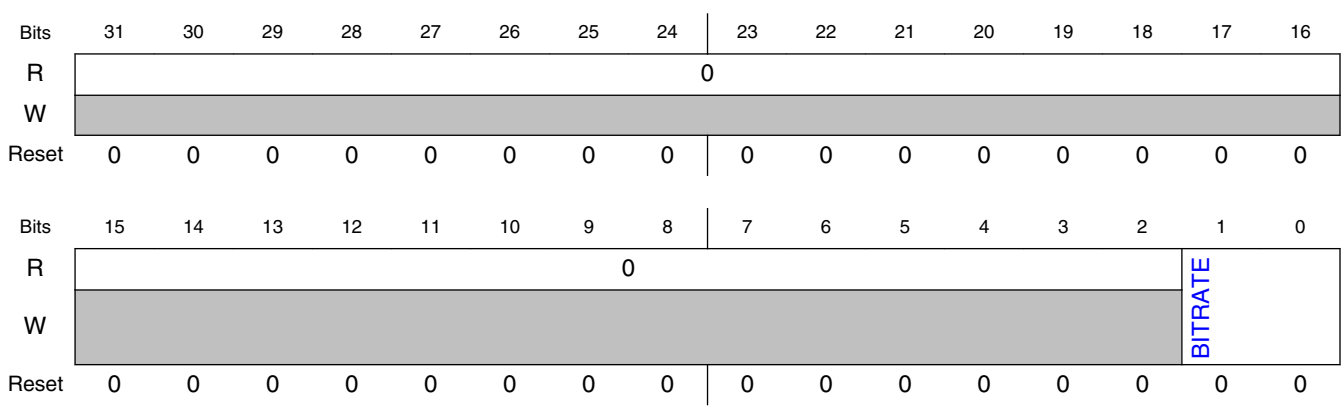
Field	Function
23 REC_BAD_PKT	Receive Bad Packets Enable Packets which fail H0-filtering, H1-filtering, or Maximum Length-filtering, to be fully received without an RX recycle (intended for debug purposes) 0b - packets which fail H0, H1, or LENGTH_MAX result in an automatic recycle after the header is received and parsed 1b - packets which fail H0, H1, or LENGTH_MAX are received in their entirety
22-16 LENGTH_MAX	Maximum Length for Received Packets Sets the Maximum Length Packet that can be received, in multiples of 16 bytes. LENGTH_MAX is compared directly against the extracted LENGTH field of the header (not the adjusted length). LENGTH_MAX=0 (default) is a special case that implies no limit.
15-12 —	Reserved.
11-0 WHITEN_SZ_THR	Whitener Size Threshold Minimum Packet Length (extracted LENGTH field) required to enable whiten. Requires WHITEN_START=2 or 3

A.3.3.30 BIT RATE (BITRATE)

A.3.3.30.1 Offset

Register	Offset
BITRATE	88h

A.3.3.30.2 Diagram



A.3.3.30.3 Fields

Field	Function
31-2 —	Reserved.
1-0 BITRATE	Bit Rate Selects the Bit Rate for the Generic FSK Link Layer. The blocks which interface to the Link Layer (e.g., PHY, CRC/Whitener), also adapt to the selected bit rate. 00b - 1Mbit/sec 01b - 500Kbit/sec 10b - 250Kbit/sec (not supported if WHITEN_CFG[MANCHESTER_EN]=1)

A.3.3.31 PACKET BUFFER PARTITION POINT (PB_PARTITION)

A.3.3.31.1 Offset

Register	Offset
PB_PARTITION	8Ch

A.3.3.31.2 Diagram

Bits	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

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

A.3.3.31.3 Fields

Field	Function
31-11 —	Reserved.
10-0	Packet Buffer Partition Point

Field	Function
PB_PARTITION	The Packet Buffer Partition Point defines the starting point for the RX segment of the Packet Buffer. The partitioning of the consolidated RAM between TX and RX is controlled by this configurable TX/RX Partition Point. PB_PARTITION can be programmed with any value between 0 (base of RAM0) and 1088 (after the last address of RAM1). The consolidated RAM is partitioned such that the TX buffer resides before the Partition Point (RAM entries 0 to (PB_PARTITION-1)), and the RX buffer resides after it (RAM entries PB_PARTITION to 1087). Programming the Partition Point register to 0 dedicates the entire consolidated RAM to RX; programming the Partition Point register to 1088 dedicates the entire consolidated RAM to TX; programming the Partition Point register to any value between 1 and 1087 yields a split TX/RX buffer. The TX segment (if any) is always first in the Packet Buffer, starting at Packet RAM address 0 of RAM0. Since the Packet Buffer RAM word-width is 2 bytes, the units for PB_PARTITION is "words". (Multiply by 2 to convert to byte addressing)

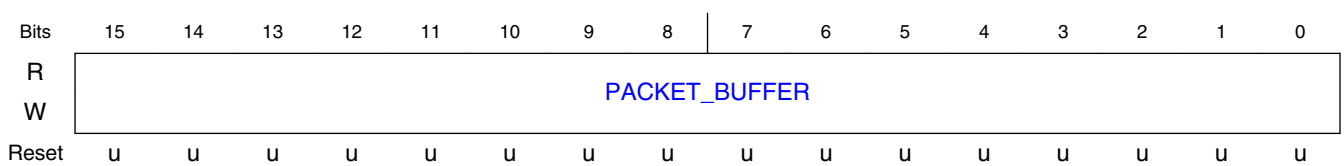
A.3.3.32 PACKET BUFFER (PACKET_BUFFER_0 - PACKET_BUFFER_1087)

A.3.3.32.1 Offset

For a = 0 to 1087:

Register	Offset
PACKET_BUFFER_a	700h + (a × 2h)

A.3.3.32.2 Diagram



A.3.3.32.3 Fields

Field	Function
15-0 PACKET_BUFFER	PACKET BUFFER RAM Storage for packet data.

Appendix B
Release Notes for Revision 5

B.1 Release Notes for Revision 5

Table B-1. Revision History

Chapter	Chapter Name	Substantial Changes
	General changes throughout the document	• Reference Manual's title changed to 'MKW36/35 Reference Manual' from 'MKW35/36 Reference Manual'. • BLE 4.2 changed to BLE 5.0 throughout.
Chapter 1	About This Manual	• No substantial changes.
Chapter 2	Introduction	• Updated Introduction section. • Updated the voltage range of DCDC in Buck configuration to 2.1 to 3.6 V from 1.8 to 4.25 V.
Chapter 3	Signal Multiplexing and Pin Assignment	• Updated signal name at pin 17 and pin 19 of 40 QFN and 48 LQFN in ALT1 functionality to PTB3/ERCLK32K.
Chapter 4	Chip Configuration	• Updated feature list for TPM0 in TPM Instantiation Information. Clarified support of Quadrature decoder and filtering in TPM0.
Chapter 5	Memory Map	• Corrected address range to 0x0008_0000 – 0x0FFF_FFFF (from 0x0008_0000 – 0x1FFE_FFFF) in System memory map.
Chapter 6	Clock Distribution	• Updated RSIM internal clock to 'OSC32KCLK' from 'ERCLK32K' in Module clocks.
Chapter 7	Reset and Boot	• No substantial changes.
Chapter 8	Power Management	• No substantial changes.
Chapter 9	Security	• No substantial changes.
Chapter 10	Debug	• No substantial changes.
Chapter 11	Kinetis Flashloader	• Added Chip-Specific Information. • Changed UART to LPUART throughout. • Added CALL command feature to the chapter. • In Memory Maps, updated the "Kinetis Flashloader RAM Memory Map" figure.
Chapter 12	Port control and interrupt (PORT)	• Updated Features section. • Added Register reset values section for PCR registers.
Chapter 13	System Integration Module (SIM)	• No substantial changes.
Chapter 14	System Mode Controller (SMC)	• No substantial changes.
Chapter 15	Power Management Controller (PMC)	• No substantial changes.

Table continues on the next page...

Table B-1. Revision History (continued)

Chapter	Chapter Name	Substantial Changes
Chapter 16	DCDC Converter (DCDC)	• Updated Functional Description related to PSWITCH voltage. • Updated values in the DCDC_VDD1P8CTRL_TRG bit description table from 0x3F 3.575 V to 0x3C 3.5 V in the DCDC_REG3 register.
Chapter 17	Low-Leakage Wakeup Unit (LLWU)	• No substantial changes.
Chapter 18	Reset Control Module (RCM)	• No substantial changes.
Chapter 19	Bit Manipulation Engine (BME)	• No substantial changes.
Chapter 20	Miscellaneous Control Module (MCM)	• No substantial changes.
Chapter 21	Micro Trace Buffer (MTB)	• No substantial changes.
Chapter 22	Crossbar Switch Lite (AXBS-Lite)	• No substantial changes.
Chapter 23	Peripheral Bridge (AIPS-Lite)	• No substantial changes.
Chapter 24	Direct Memory Access Multiplexer (DMAMUX)	• No substantial changes.
Chapter 25	DMA Controller Module	• Compressed display of memory map definition for TCD register arrays in DMA register descriptions. • Added the following note in SSIZE bit description: "The eDMA defaults to privileged data access for all transactions." • Changed section heading from "Channel n Priority Register" to "Channel Priority Register" in Channel Priority Register (DCHPRI0 - DCHPRI3). Also removed a note about reset value from CHPRI bit description. • Added a note to BWC bit description. • Changed Sigma Delta Analog to Digital Convertor (SDADC) to ADC in Suspend/resume a DMA channel with active hardware service requests
Chapter 26	Multipurpose Clock Generator (MCG)	• No substantial changes.
Chapter 27	32 kHz Oscillator (32kRTC)	• No substantial changes.
Chapter 28	Flash Memory Controller (FMC)	• No substantial changes.
Chapter 29	Flash Memory Module (FTFE)	• Added FAC Application Tips topic. • Updated Program Flash Swap IFR address range to 0x04_0000 - 0x04_03FF in Table 29-18. • Corrected shift amount when storing Swap Indicator Address in the Program Flash Swap IFR during SWAP initialization in Swap Control command (program flash only devices).
Chapter 30	Analog-to-digital converter (ADC)	• No substantial changes.
Chapter 31	Comparator (CMP)	• No substantial changes.
Chapter 32	Voltage Reference(VREF)	• No substantial changes.
Chapter 33	Timer/PWM Module (TPM)	• No substantial changes.

Table continues on the next page...

Table B-1. Revision History (continued)

Chapter	Chapter Name	Substantial Changes
Chapter 34	Periodic interrupt timer (PIT)	• Changed the display of offset addresses of registers. • Added a note in the topic Timers related to timer pause.
Chapter 35	Low-power timer (LPTMR)	• No substantial changes.
Chapter 36	Real Time Clock (RTC)	• No substantial changes.
Chapter 37	FlexCAN	• In Introduction, changed word "Chip" to "FlexCAN" in figure "FlexCAN block diagram". • Added notes about referring to the CAN Protocol standard (ISO 11898-1) in the memory map/register definition section. • Added the following note in FlexCAN memory mapping : "An invalid register access will result in a bus error. This includes reading a write-only register, writing a read-only register or accessing an invalid address." • Added a note to the bit description of CTRL1[CLKSRC] about referring to the clock distribution chapter (module clocks table) to identify the proper clock source. • In Error Counter (ECR), changed access type of fields ECR[RXERRCNT_FAST] and ECR[TXERRCNT_FAST] from read-write to read-only-writes-zero. • Updated Transmit process for increased clarity. • In Transmission abort mechanism, removed summary of procedure at end. • Added Table 37-20. • In Transceiver Delay Compensation, added new text, equation and figure to provide more information about how to use CAN_FDCTRL[TDCOFF]. Also removed phrase "(e.g. half of the bit time in the data phase)" from paragraph that precedes the updated content. In addition, changed the value in the sentence "The TDCVAL value saturates..." from 15 to 63. • In Protocol timing, added a note about referring to the clock distribution chapter (module clocks table) to identify the proper clock source.
Chapter 38	Serial Peripheral Interface (SPI)	• Updated Status Register SR[16] bit to ROZ read-only zero.
Chapter 39	Inter-Integrated Circuit (I2C)	• No substantial changes.
Chapter 40	Low power Universal Asynchronous Receiver/Transmitter (LPUART)	• Updated Features to add FIFO and Watermark support. • Added FIFO and WATER registers. Refer LPUART Register definition.
Chapter 41	Carrier Modulator Transmitter (CMT)	• No substantial changes.
Chapter 42	General-purpose input/ output (GPIO)	• No substantial changes.
Chapter 43	LP Trusted Cryptography (LTC)	• No substantial changes.
Chapter 44	True Random Number Generator (TRNG)	• Changed TRNG_MCTL[LRUN_CONT] at offset 14 to Reserved.
Chapter 45	2.4 GHz Multi-Protocol Radio	• In the 2.4GHZ RADIO V2 TSM block diagrams of the topics, Block diagram and TSM-Controlled Outputs, changed *sy_vco_autotune_en* signal to *pll_loop_is_open*. • Updated descriptions of RECYCLE_COUNT2, RECYCLE_COUNT1, and RECYCLE_COUNT0 bits in the TSM RECYCLE COUNT (RECYCLE_COUNT) register. • In Platform Register descriptions for the Bluetooth Link Layer,

Table continues on the next page...

Table B-1. Revision History (continued)

Chapter	Chapter Name	Substantial Changes

		• updated bits 8 and 9 in the Event Status Register table, Table 45-37 as follows:
		  • Bit 8 = DTM_INTR: (DTM is on) AND (RX packet done)
		  • Bit 9 = (TSM_IRQ) OR (DTM TX complete)
		• added *wl_clk_gate_en* field at bit 15 in the DSM configuration register table, Table 45-37.
		• Added the following method to the list of methods of enabling the RF Osc in the topic RF Osc Enable : "The external pin can be asserted to request RF Osc be enabled.".
		• Added RF Osc Debug Visibility topic to RSIM module.
		• Added Radio Miscellaneous (MISC) register to RSIM Memory map.
		• Updated description of XTAL_OUT_BUF_EN bit in the Radio Analog Test Registers (ANA_TEST) register XTAL_OUT_BUF_EN.
		• Updated description of CS_FAIL bit in the LOCK_DETECT register.
		• Added a note in Transceiver Memory Map and Register Definition about XCVR_CTRL and XCVR_MISC register names used as aliases in the XCVR_MISC module.
		• Added DIGITAL TEST MUX CONTROL (DTEST_CTRL) register to XCVR_CTRL_ADDR Memory map.
		• Removed External Signal Descriptions section from Transceiver DMA and Packet RAM Debug Modes chapter.
		• Updated description of DBG_RAM_FULL bit in the PACKET RAM CONTROL (PACKET_RAM_CTRL) register.

Appendix A	Radio Register Summary	• Updated Appendix title from "Radio Register Overview" to "Radio Register Summary". • Updated description of VCO_DAC_REF_ADJUST bit in RX_AUXPLL register.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Document Number MKW36A512RM
Revision 5, 07/2018

Preliminary