S32K144 Reference Manual

For Assessment Purposes Only

Document Number: S32K144RM
Rev. 1 Draft H, 02/2016

Review Draft
Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Contents

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

Chapter 1
About This Manual

Chapter 2
Introduction

For Assessment Purposes Only

Chapter 3
Core Overview

Chapter 4
Memories and Memory Interfaces

For Assessment Purposes Only

Chapter 5
Memory Map

Chapter 6
Clock Distribution

For Assessment Purposes Only

Chapter 7
Reset and Boot

Chapter 8
Power Management

For Assessment Purposes Only

Chapter 9
Security

Chapter 10
Debug

For Assessment Purposes Only

Chapter 11
Signal Multiplexing and Pin Assignment

Chapter 12
Port Control and Interrupts (PORT)

For Assessment Purposes Only

Chapter 13
System Integration Module (SIM)

For Assessment Purposes Only

Chapter 14
Reset Control Module (RCM)

Chapter 15
System Mode Controller (SMC)

For Assessment Purposes Only

Chapter 16
Power Management Controller (PMC)

Chapter 17
Miscellaneous Control Module (MCM)

For Assessment Purposes Only

Chapter 18
Crossbar Switch Lite (AXBS-Lite)

Chapter 19
Memory Protection Unit (MPU)

For Assessment Purposes Only

Chapter 20
AIPS-Lite

For Assessment Purposes Only

Chapter 21
Direct Memory Access Multiplexer (DMAMUX)

Chapter 22
Enhanced Direct Memory Access (eDMA)

For Assessment Purposes Only

Chapter 23
Trigger MUX Control (TRGMUX)

Chapter 24
External Watchdog Monitor (EWM)

For Assessment Purposes Only

Chapter 25
Error Injection Module (EIM)

For Assessment Purposes Only

Chapter 26
Error Reporting Module (ERM)

Chapter 27
Watchdog timer (WDOG)

For Assessment Purposes Only

Chapter 28
System Clock Generator (SCG)

For Assessment Purposes Only

Chapter 29
Peripheral Clock Controller (PCC)

For Assessment Purposes Only

Chapter 30
Local Memory Controller (LMEM)

For Assessment Purposes Only

Chapter 31
Miscellaneous System Control Module (MSCM)

For Assessment Purposes Only

Chapter 32
Flash Memory Controller (FMC)

Chapter 33
Flash Memory Module (FTFE)

For Assessment Purposes Only

Chapter 34
Cyclic Redundancy Check (CRC)

Chapter 35
Analog-to-Digital Converter (ADC)

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 36
Comparator (CMP)

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 37
Programmable Delay Block (PDB)

For Assessment Purposes Only

Chapter 38
FlexTimer Module (FTM)

For Assessment Purposes Only

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 39
Low Power Interrupt Timer (LPIT)

For Assessment Purposes Only

Chapter 40
Low Power Timer (LPTMR)

For Assessment Purposes Only

Chapter 41
Real Time Clock (RTC)

Chapter 42
Low Power Serial Peripheral Interface (LPSPI)

For Assessment Purposes Only

Chapter 43
Low Power Inter-Integrated Circuit (LPI2C)

For Assessment Purposes Only

Chapter 44
Low Power Universal Asynchronous Receiver/Transmitter (LPUART)

Chapter 45
Flexible I/O (FlexIO)

For Assessment Purposes Only

Chapter 46
FlexCAN

For Assessment Purposes Only

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 47
General-Purpose Input/Output (GPIO)

Chapter 48
JTAG Controller (JTAGC)

For Assessment Purposes Only

For Assessment Purposes Only

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.

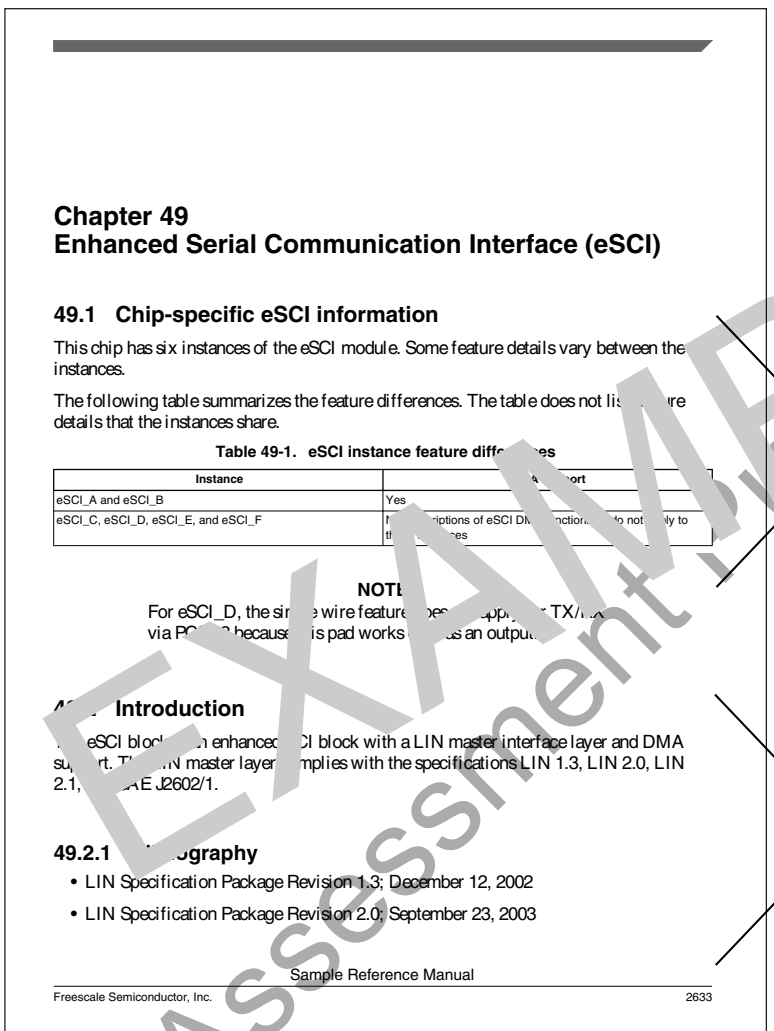
1. Chapters in the first set contain information that applies to all components on the chip.
2. Chapters in the second set are organized into functional groupings that detail particular areas of functionality.
   - Examples of these groupings are clocking, timers, and communication interfaces.
   - Each grouping includes chapters that provide a technical description of individual modules.
3. **Spreadsheets attached to this document**
   - S32K144_IO_Signal_Description_Input_Multiplexing.xlsx
   - S32K1xx_memory_map.xlsx
   - DMA_Interrupt_mapping.xlsm

1.3 Module descriptions

Each module chapter has two main parts:

For Assessment Purposes Only

- The first section, *Chip-specific [module name] information*, provides details such as the number of module instances on the chip and connections between the module and other modules. Read this section *first* because its content is crucial to understanding the information in other sections of the chapter.
- The subsequent sections provide general information about the module, including its signals, registers, and functional description.



Chapter 49
Enhanced Serial Communication Interface (eSCI)

49.1 Chip-specific eSCI information

This chip has six instances of the eSCI module. Some feature details vary between the instances.

The following table summarizes the feature differences. The table does not li re details that the instances share.

Table 49-1. eSCI instance feature diff es

Instance	ort
eSCI_A and eSCI_B	Yes
eSCI_C, eSCI_D, eSCI_E, and eSCI_F	riptions of eSCI DM nctio o not ly to th es

NOTE
For eSCI_D, the si e wire feature oes pp TX/ via PC because is pad works as an output

Introduction

eSCI blo n enhanced CI block with a LIN master interface layer and DMA su rt. T N master layer mplies with the specifications LIN 1.3, LIN 2.0, LIN 2.1, AE J2602/1.

49.2.1 ography

- LIN Specification Package Revision 1.3; December 12, 2002
- LIN Specification Package Revision 2.0; September 23, 2003

Sample Reference Manual
Freescale Semiconductor, Inc. 2633

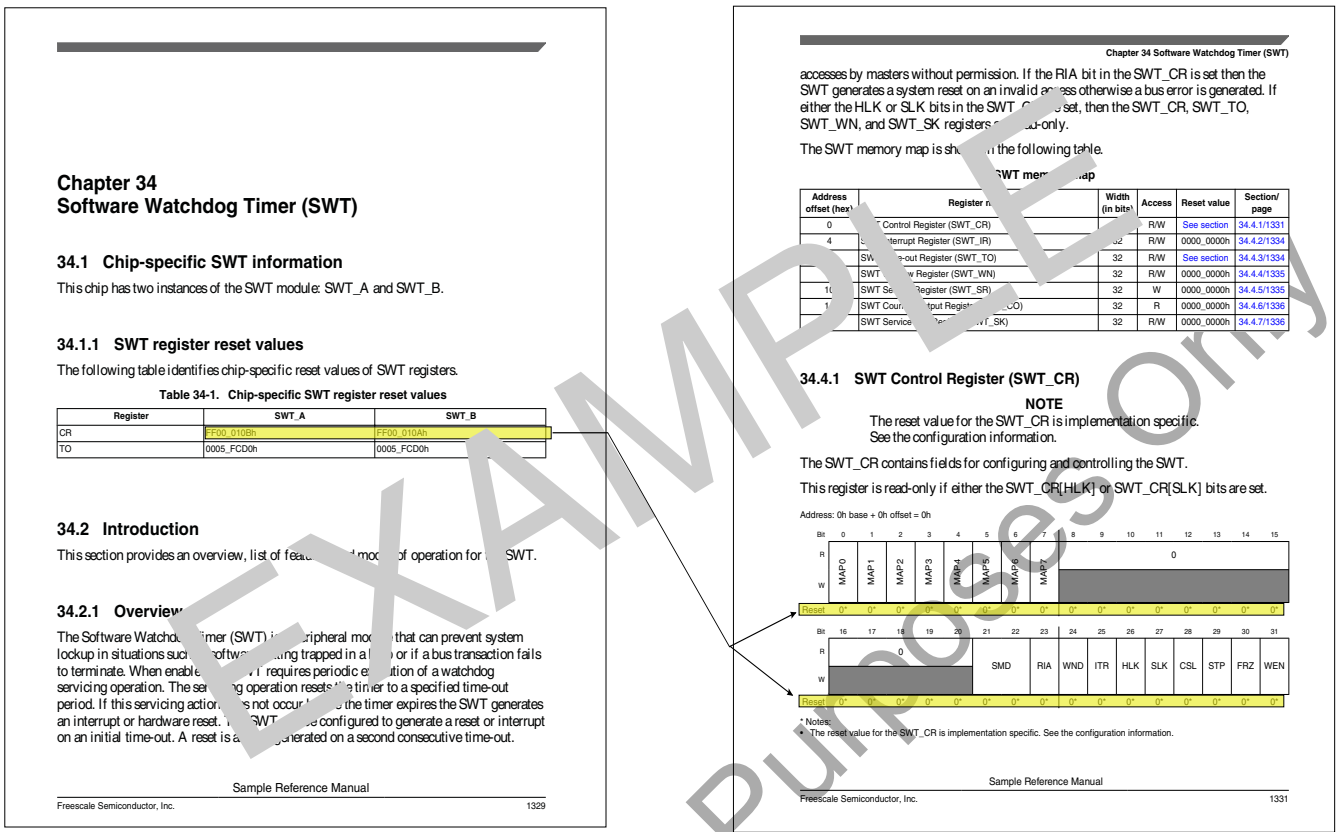
Figure 1-1. Example: chapter chip-specific information and general module information

EXAMPLE
For Assessment Purposes Only

1.3.1 Example: chip-specific information that clarifies content in the same chapter

The example below shows chip-specific information that clarifies general module information presented later in the chapter. In this case, the chip-specific register reset values supercede the reset values that appear in the register diagram.

Freescale Confidential Proprietary - Non-Disclosure Agreement required



Chapter 34
Software Watchdog Timer (SWT)

34.1 Chip-specific SWT information

This chip has two instances of the SWT module: SWT_A and SWT_B.

34.1.1 SWT register reset values

The following table identifies chip-specific reset values of SWT registers.

Table 34-1. Chip-specific SWT register reset values

Register	SWT_A	SWT_B
CR	FF00_010Bh	FF00_010Ah
TO	0005_FCD0h	0005_FCD0h

34.2 Introduction

This section provides an overview, list of fea... mo... of operation for ... SWT.

34.2.1 Overview

The Software Watchdo... imer (SWT) i... ripheral mo... that can prevent system lockup in situations suc... softw... ting trapped in a l... or if a bus transaction fails to terminate. When enable... requires periodic e... ution of a watchdog servicing operation. The se... g operation resets ... timer to a specified time-out period. If this servicing actio... s not occur ... the timer expires the SWT generates an interrupt or hardware reset. ... SWT ... e configured to generate a reset or interrupt on an initial time-out. A reset is a... generated on a second consecutive time-out.

Sample Reference Manual

Freescale Semiconductor, Inc. 1329

Chapter 34 Software Watchdog Timer (SWT)

accesses by masters without permission. If the RIA bit in the SWT_CR is set then the SWT generates a system reset on an invalid a... ss otherwise a bus error is generated. If either the HLK or SLK bits in the SWT_C... set, then the SWT_CR, SWT_TO, SWT_WN, and SWT_SK registers ... -only.

The SWT memory map is sh... n the following table.

SWT mem... ap

Address offset (hex)	Register n...	Width (in bits)	Access	Reset value	Section/ page
0	... Control Register (SWT_CR)	...	R/W	See section	34.4.1/1331
4	... terrupt Register (SWT_IR)	...	R/W	0000_0000h	34.4.2/1334
...	SW... e-out Register (SWT_TO)	32	R/W	See section	34.4.3/1334
...	SWT ... w Register (SWT_WN)	32	R/W	0000_0000h	34.4.4/1335
10	SWT Se... Register (SWT_SR)	32	W	0000_0000h	34.4.5/1335
1...	SWT Cour... tput Regist... CO)	32	R	0000_0000h	34.4.6/1336
...	SWT Service ... (SWT_SK)	32	R/W	0000_0000h	34.4.7/1336

34.4.1 SWT Control Register (SWT_CR)

NOTE
The reset value for the SWT_CR is implementation specific.
See the configuration information.

The SWT_CR contains fields for configuring and controlling the SWT.

This register is read-only if either the SWT_CR[HLK] or SWT_CR[SLK] bits are set.

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

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	MAP0	MAP1	MAP2	MAP3	MAP4	MAP5	MAP6	MAP7	0							
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0					SMD	RIA	WND	ITR	HLK	SLK	CSL	STP	FRZ	WEN	
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- The reset value for the SWT_CR is implementation specific. See the configuration information.

Sample Reference Manual

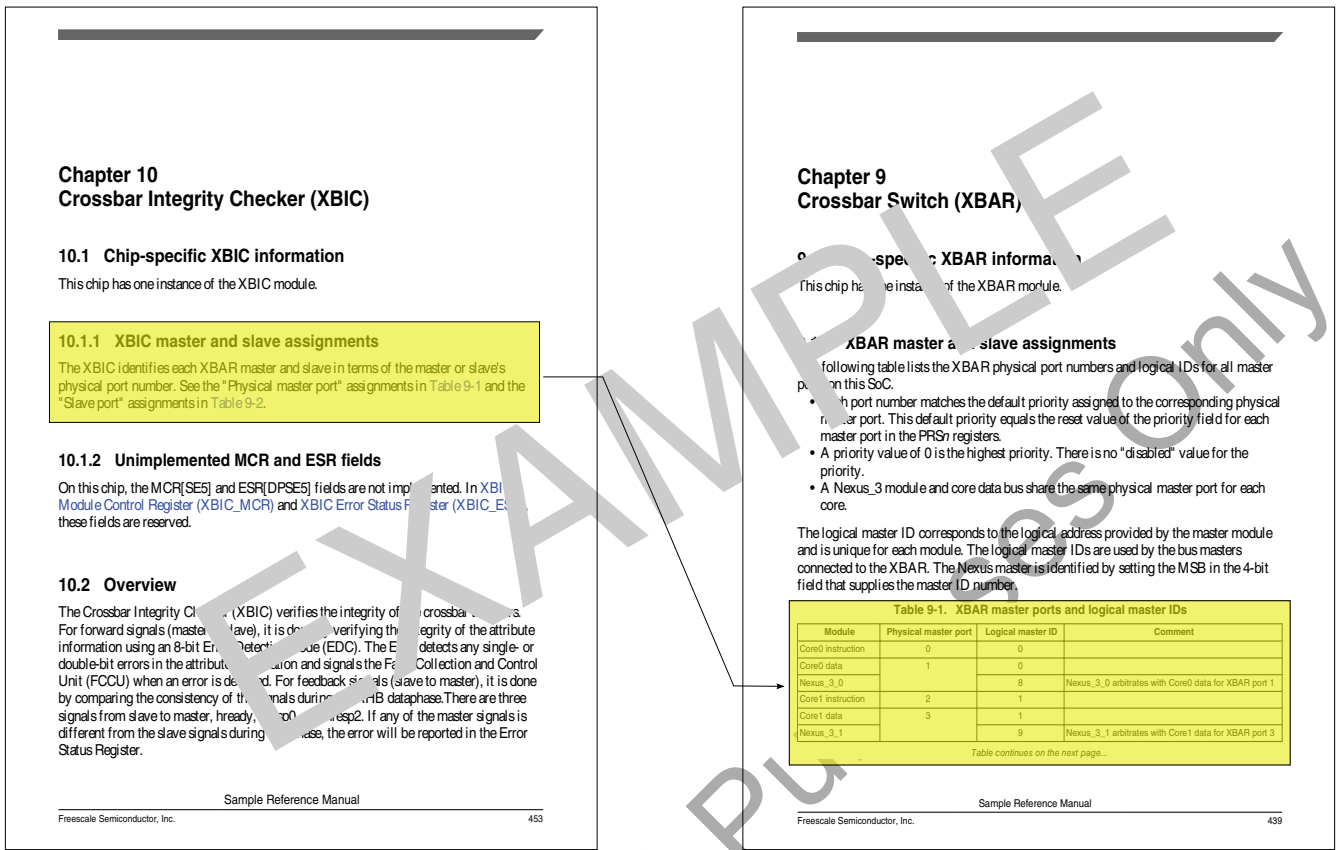
Freescale Semiconductor, Inc. 1331

Figure 1-2. Example: chip-specific information that clarifies content in the same chapter

1.3.2 Example: chip-specific information that refers to a different chapter

The chip-specific information below refers to another chapter's chip-specific information. In this case, read both sets of chip-specific information before reading further in the chapter.

EXAMPLE
For Assessment Purposes Only



Chapter 10
Crossbar Integrity Checker (XBIC)

10.1 Chip-specific XBIC information

This chip has one instance of the XBIC module.

10.1.1 XBIC master and slave assignments

The XBIC identifies each XBAR master and slave in terms of the master or slave's physical port number. See the "Physical master port" assignments in Table 9-1 and the "Slave port" assignments in Table 9-2.

10.1.2 Unimplemented MCR and ESR fields

On this chip, the MCR[SE5] and ESR[DPSE5] fields are not implemented. In XBI Module Control Register (XBIC_MCR) and XBIC Error Status Register (XBIC_ES these fields are reserved.

10.2 Overview

The Crossbar Integrity Ch... (XBIC) verifies the integrity of ... crossbar ... s. For forward signals (maste... lave), it is do... verifying th... egrity of the attribute information using an 8-bit Er... Detecti... ue (EDC). The E... detects any single- or double-bit errors in the attribut... ation and signals the Fa... Collection and Control Unit (FCCU) when an error is de... d. For feedback si...als (slave to master), it is done by comparing the consistency of th... nals duri... HB dataphase.There are three signals from slave to master, hready, hr... resp2. If any of the master signals is different from the slave signals during ... ase, the error will be reported in the Error Status Register.

Sample Reference Manual

Freescale Semiconductor, Inc. 453

Chapter 9
Crossbar Switch (XBAR)

9... -spe... : XBAR informa...

This chip ha... e insta... of the XBAR module.

... XBAR master a... slave assignments

The following table lists the XBAR physical port numbers and logical IDs for all master p... on this SoC.

- ... h port number matches the default priority assigned to the corresponding physical m... er port. This default priority equals the reset value of the priority field for each master port in the PRSn registers.
- A priority value of 0 is the highest priority. There is no "disabled" value for the priority.
- A Nexus_3 module and core data bus share the same physical master port for each core.

The logical master ID corresponds to the logical address provided by the master module and is unique for each module. The logical master IDs are used by the bus masters connected to the XBAR. The Nexus master is identified by setting the MSB in the 4-bit field that supplies the master ID number.

Table 9-1. XBAR master ports and logical master IDs

Module	Physical master port	Logical master ID	Comment
Core0 instruction	0	0	
Core0 data	1	0	
Nexus_3_0		8	Nexus_3_0 arbitrates with Core0 data for XBAR port 1
Core1 instruction	2	1	
Core1 data	3	1	
Nexus_3_1		9	Nexus_3_1 arbitrates with Core1 data for XBAR port 3

Table continues on the next page...

Sample Reference Manual

Freescale Semiconductor, Inc. 439

Figure 1-3. Example: chip-specific information that refers to a different chapter

1.4 Register descriptions

Module chapters present register information in:

- Memory maps containing:
  - An offset from the module's base address
  - 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.

EXAMPLE
For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

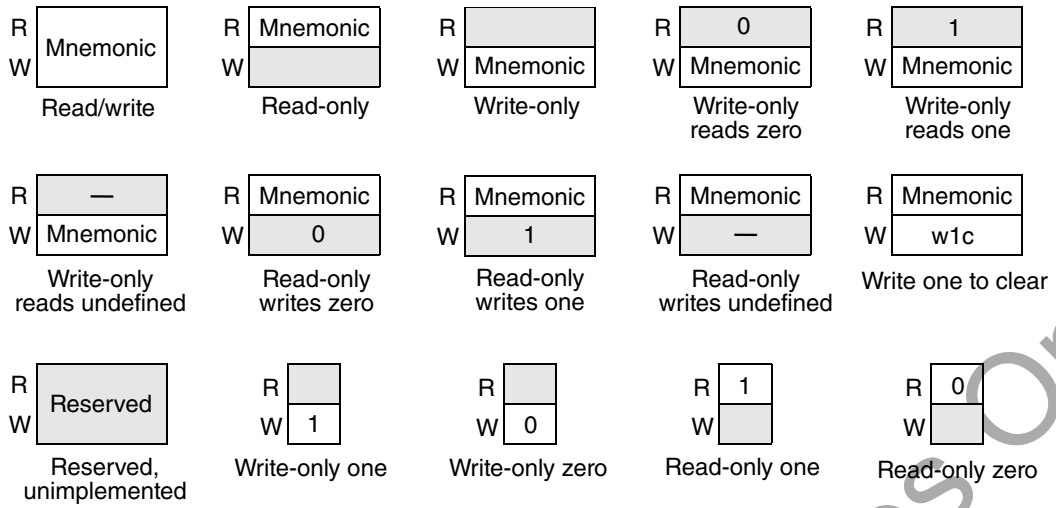


Figure 1-4. Register figure conventions

1.5 Conventions

1.5.1 Notes, Cautions, and Warnings

Note, Caution, and Warning notices appear throughout this manual. Each notice type alerts readers to a specific kind of information.

NOTE

Notes convey information that may be tangential to a topic or that may not apply to all readers.

CAUTION

Caution notices call out information that readers should know before taking further action. Cautions frequently point to trouble spots that may damage the chip or board.

WARNING

Warning notices inform readers about actions that could result in unwanted consequences, especially those that may cause bodily injury.

1.5.2 Numbering systems

The following suffixes identify different numbering systems:

For Assessment Purposes Only

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*.

1.5.3 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.

1.5.4 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...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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."

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 2
Introduction

2.1 Overview

The S32K1xx Family further extends the highly scalable portfolio of ARM® Cortex®-M0+/M4F MCUs in the automotive industry. It builds on the legacy of the KEA series, whilst introducing higher memory options alongside a richer peripheral set extending capability into a variety of automotive applications. With a 2.70 –5.5 V supply and focus on automotive environment robustness, the S32K series devices are well suited to a wide range of applications in electrical harsh environments, and are optimized for cost-sensitive applications offering low pin-count options. The S32K series offers a broad range of memory, peripherals, and package options. They share common peripherals and pin counts allowing developers to migrate easily within an MCU family or among the MCU families to take advantage of more memory or feature integration. This scalability allows developers to standardise on the S32K series for their end product platforms, maximising hardware and software reuse and reducing time-to-market.

2.2 Device Introduction

This device is a 32-bit general purpose automotive microcontroller based on ARM Cortex-M4F core. It offers superior performance, large memories and the most scalable peripherals in this class. This product series provides up to 112MHz CPU performance with DSP and FPU support, up to 512KB Flash and 64KB SRAM. The overview of the device features are as follows:

- 32-bit ARM Cortex-M4F core with FPU, up to 112 MHz (HSRUN) and 80 MHz (Normal RUN)
- 512 KB Code Flash and 64 KB FlexMem (support up to 4 KB Emulated EEPROM with 4 KB FlexRAM)
- 64 KB RAM supporting both CPU private access and crossbar access to provide parallel access of instruction and data

For Assessment Purposes Only

- Modified Harvard connections with Local Memory controller (LMEM) to support tightly coupled RAM and 4 KB I/D cache
- Integrated clocking architecture with on-chip Fast IRC 48 MHz, Slow IRC 8MHz / 2 MHz, 128 KHz LPO and a PLL unit.
- Analog modules to provide precision mixed-signal capabilities, include 12-bit 1 Msps SAR ADC, high-speed Comparator etc.
- Power Management Controller (PMC) with internal regulators capable of supporting multiple power modes which include HSRUN, RUN, WAIT, STOP, VLPR, VLPW and VLPS.
- IO to support 2.7V to 5.5V supply
- Wide operating voltage ranges from 2.7–5.5 V with fully functional flash program/ erase/read operations
- 64LQFP, 100LQFP and 100 MAPBGA packages with up to 89 GPIO pins
- Ambient operating temperature ranges from –40 °C to 125 °C

The S32K14x MCU portfolio is supported by the most-comprehensive set of development tools and software. The enablement package includes NXP Arduino compatible evaluation boards, S32K Software Development Kit (SDK) including graphical configurability and S32 Design Studio software, as well as broad support from IAR Systems, Cosmic Software, Green Hills and other partners.

ment Purposes Only

2.3 Feature Summary

The following table lists the features integrated on this device.

Table 2-1. Device feature summary

Feature	Summary for this device
Hardware Characteristics	
Package	64-pin LQFP, 10 mm× 10 mm, 0.5mm pitch 100-pin LQFP, 14 mm× 14 mm, 0.5 mm pitch 100-pin MAPBGA, 11 mm x 11 mm, 1 mm ball pitch
Voltage range	2.7 V to 5.5 V
Temperature range (T_A)	-40°C to 125°C
Temperature range (T_J)	-40°C to 135°C
System	
Central processing unit (CPU)	ARM Cortex-M4F
Max. CPU frequency	112MHz
Digital signal processor (DSP)	Yes
Floating point unit (FPU)	Yes
Memory protection unit (MPU)	Yes

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 2-1. Device feature summary (continued)

Feature	Summary for this device
I/D Cache	4 KB I/D cache
Nested vectored Interrupt controller (NVIC)	up to 240 vectored interrupts 16 programmable interrupt priority levels
Wake-up interrupt controller (WIC)	Yes
Direct memory access (DMA)	16 channels
DMA request multiplex (DMAMUX)	Yes
Non-maskable interrupt (NMI)	Yes
Software watchdog	Yes
Hardware watchdog	Yes, with external monitor pin.
Debug	2-pin serial wire debug (SWD) IEEE 1149.1 Joint Test Access Group (JTAG)
Trace	Trace Port Interface Unit (TPIU)
Boundary scan	Yes
Unique Identification (ID) Number	128-bit wide
Memory	
Program Flash memory	512 KB (2x 256 KB array)
FlexMemory	64 KB Data flash (D-flash)/EEPROM backup (E-Flash) memory: 4 KB additional FlexRAM supporting high-endurance, non-volatile EEPROM
Flash cache	No (single speculative prefetch buffer only)
Flash Access Control (FAC)	No
Random-access memory (RAM)	64 KB[1]
Low-leakage standby memory	RAM retained in all modes
Cyclic redundancy check (CRC)	16- or 32-bit CRC with programmable generator polynomial
Clocks	
System clock generator (SCG)	SCG module with integrated OSC, FIRC, SIRC, PLL
External crystal oscillator or resonator	OSC: 4 - 40 MHz with low power or full-swing
External square wave input clock	DC to 50 MHz
Internal clock references (IRC)	48 MHz internal IRC (FIRC) with 1% max deviation across full temperature 8MHz / 2MHz internal IRC (SIRC) with 3% max deviation across full temperature 128 kHz low power oscillator (LPO)
Phase-locked loop (PLL)	up to 360 MHz VCO
Human-Machine Interface (HMI)	
General-purpose input/output (GPIO)	Up to 89 GPIOs Pin interrupt / DMA request capability Configurable digital glitch filter

Table continues on the next page...

Table 2-1. Device feature summary (continued)

Feature	Summary for this device
	Hysteresis, pull up device and pull down device on all input pins Passive input filter on NMI_b and RESET_B input pins only High drive configurable option on 11 pins (PTA10, PTB4, PTB5, PTB6, PTD0, PTD1, PTD15, PTD16, PTE0, PTE1, and PTE4).
Segment liquid crystal display (SLCD)	No
Analog	
Power management controller (PMC)	Low voltage warning Overdrive regulation range 128 kHz LPO clock
12-bit analog-to-digital converter (ADC0–ADC1)	1 Msps with 12-bit mode 1.2 Msps with 10-bit mode 16 single-ended external channels 16 control and result registers support result compare
High-speed comparator (CMP)	x1
Timers	
Programmable delay block 0 (PDB0)	2 ADC channel with 8 pre-triggers for each channel for ADC0 1 pulse-out channel
Programmable delay block 1 (PDB1)	2 ADC channel with 8 pre-triggers for each channel for ADC1 1 pulse-out channel
Flexible timer 0 (FTM0)	16-bit, 8 channels FTM function with GTB and Global Load No Quadrature Decoder
Flexible timer 1 (FTM1)	16-bit, 8 channels FTM function with GTB and Global Load Quadrature Decoder
Flexible timer 2 (FTM2)	16-bit, 8 channels FTM function with GTB and Global Load Quadrature Decoder
Flexible timer 3 (FTM3)	16-bit, 8 channles FTM function with GTB and Global Load No Quadrature Decoder
32-bit Low-power programmable interrupt timer (LPIT)	1x LPIT 4 independent channels
Real-time clock (RTC)	Yes
Low-power timer (LPTMR)	1x LPTMR 1-channel, 16-bit pulse counter or Periodic interrupt

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 2-1. Device feature summary (continued)

Feature	Summary for this device
Communication Interfaces	
Control Area Network (CAN)	3x FlexCAN (CAN0-supports ISO CAN-FD) CAN0 supports 32 Message Buffers (MBs); CAN1 or CAN2 each supports up to 16 Message Buffers (MBs)
Low Power Serial peripheral interface (LPSPI0–LPSPI2)	3x LPSPI Slave mode functional in Stop/VLPS DMA support, 4 word FIFO support on all LPSPIs
Low Power Inter-Integrated Circuit (LPI^{2}C0)	1x LPI^{2}C Standard SMBUS compatible I^{2}C 4 word FIFO support on LPI^{2}C0 DMA support 1 Mbps ability (even with max. I^{2}C bus loading - 400pF) **Note:** only high-drive pins are able to support 1 Mbps
Low Power UART (LPUART0–LPUART2)	3x LPUART Standard features Configurable from 4x to 32x oversampling Functional in STOP/VLPS modes LIN slave operation support 32bit data width Support DMA, 4 word FIFO support on all LPUARTs
FlexIO	1x FlexIO 4 shifters, 4 timers, 8 pins Support UART, I2C, I2S, SPI and PWM generation

1. 4KB is part of EEERAM solution

For Assess

2.4 Block Diagram

The following figure shows a S32K1xx Family block diagram.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

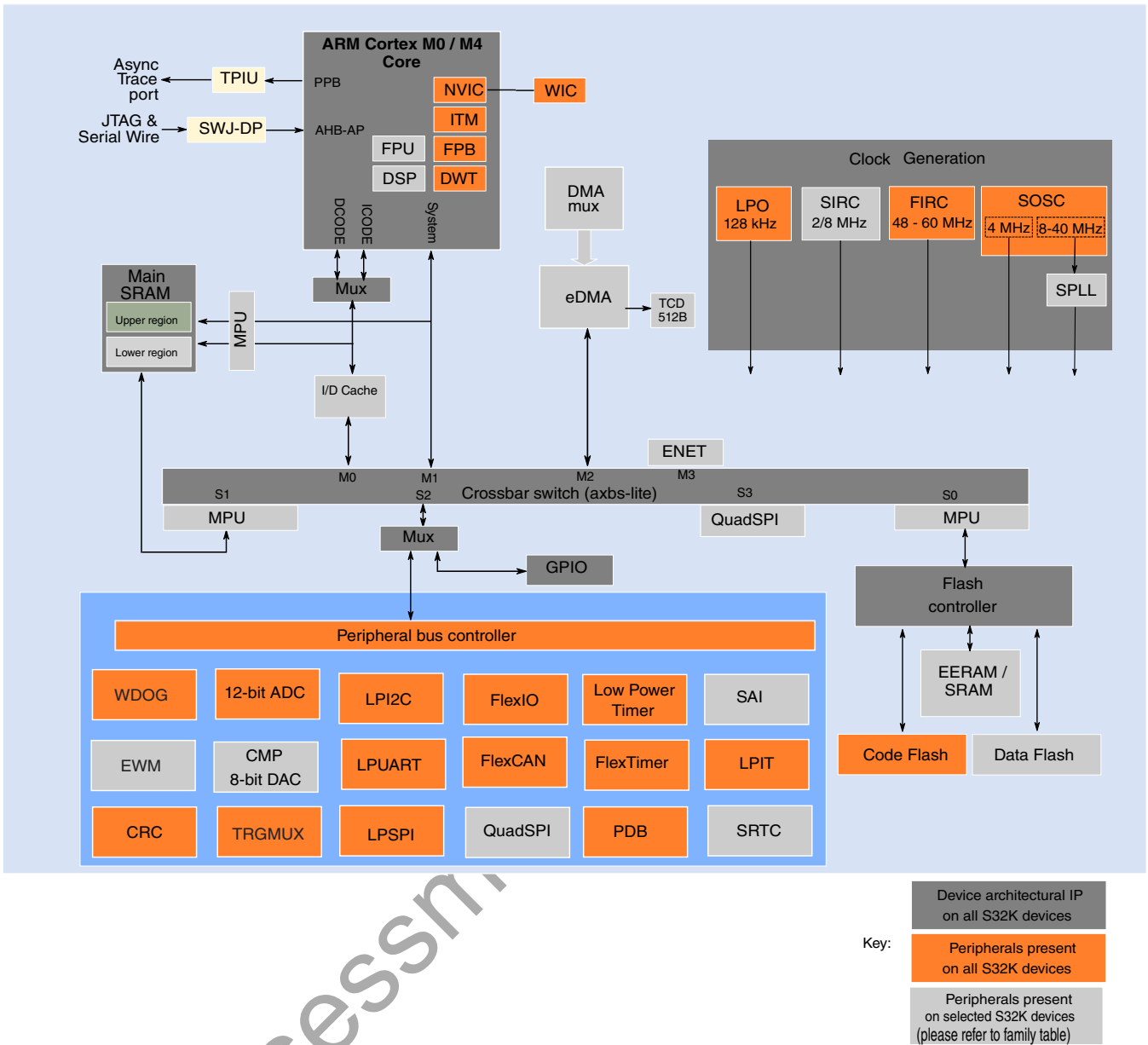


Figure 2-1. S32K1xx Family block diagram

For Assessment

2.5 Feature comparison

The following figure summarizes the memory and package options for the S32K series and demonstrates where this device fits within the overall series. All devices which share a common package are pin-to-pin compatible.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

		S32K11x			S32K14x			
	Parameter	K114	K116	K118	K142	K144	K146	K148
System	Core	ARM® Cortex™-M0+			ARM® Cortex™-M4F			
	Frequency	48 MHz	64 MHz		80 MHz	112 MHz		
	IEEE-754 FPU	o			●			
	HW Security Module (SHE)	o		●	● (programmable by NXP)			
	CRC Module	●			●			
	ISO 26262	capable up to ASIL-B			capable up to ASIL-B			
	Peripheral Speed	up to 48 MHz	up to 64 MHz		up to 80 MHz	up to 112 MHz		
	Crossbar (Concurrent Access)	●			●			
	DMA	o		●	●			
	Memory Protection Unit	●			●			
	Watchdog	●			●			
	Real-Time Clock	●			●			
	Low Power Domain	●			●			
	Number of I/Os	up to 42	up to 58		up to 89	up to 89	up to 128	up to 152
	Single Supply Voltage	2.7 - 5.5V			2.7 - 5.5V			
	Operating Temperature (T_a) Temperature Ambient	-40 to +85ºC / +105ºC / +125ºC			-40 to +85ºC / +105ºC / +125ºC			
Memory	Flash	64 KB	128 KB	256 KB	256 KB	512 KB	1 MB	2 MB
	Error Correction Code (ECC)	●			●			
	System RAM	6 KB	14 KB	22 KB	28 KB	60 KB	124 KB	252 KB
	Flex RAM	2 KB			4 KB			
	Cache	o			4 KB			
	EEPROM	1 KB	2 KB		4 KB (up to 64 KB D-Flash)		4 KB (up to 512 KB D-Flash)	
	External Memory Interface	o			o			QuadSPI
Timer	Low Power Interrupt Timer	1x			1x			
	FlexTimer (16-bit counter) 8 channels	1x (8)		2x (16)	4x (32)	4x (32)	6x (48)	8x (64)
	Programmable ADC Trigger	o			2x			
Analog	Complex Trigger Mux	1x (16)			1x (64)			
	12-bit SAR ADC (1 MSPS each)	1x (12)	1x (16)		2x (16)		2x (24)	2x (32)
	Comparator with 8-bit DAC	●			●			
Communication	100 Mbit IEEE-1588 Ethernet MAC	o			o			1x
	Serial Audio Interface (AC97, TDM, I2S)	o			o			1x
	Low Power UART/LIN	1x			2x	3x		
	Low Power SPI	1x		2x	2x	3x		
	Low Power I2C	1x			1x			2x
	FlexCAN (CAN-FD ISO/CD 11898-1)	1x (1x with FD)			2x (1x with FD)	3x (1x with FD)	3x (2x with FD)	3x (3x with FD)
	FlexIO (configurable # of UART, SPI, I2C, I2S)	1x			1x			
Applications	Touch Interface	●			●			
	Wireless Charging	o			●			
	Audio Streaming	o			o			●
	Over The Air Firmware Update*	o			o			●
	BLDC / PMSM Motor Control	o			●			
	Ethernet Connected Edge node with support for BroadR-Reach®	o			o			●
	Audio Streaming over Ethernet	o			o			●
	Secure data transmission over Ethernet includes VLAN support and ISO CAN-FD with security engine support	o			o			●
	32 TDM channel serial audio streaming (in and output) for radio, amplifier and microphone applications	o			o			●
	Large 256 KB internal SRAM enabling usage of standard TCP/IP stacks	o			o			●
	CAN to CAN or Ethernet to CAN firewall	o			●			
	External Memory extension by a dedicated QuadSPI interface, e.g. for hand writing recognition applications	o			o			●
	Human Machine Interface Applications	o			●			
IDEs	Debug & Trace	SWD			SWD, JTAG (ITM, SWV, SWO)			+ETM
	Ecosystem (IDE, Compiler, Debugger)	NXP S32 Design Studio (GCC) + SDK, IAR, GHS, COSMIC, Lauterbach, iSystems			NXP S32 Design Studio (GCC) + SDK, IAR, GHS, COSMIC, Lauterbach, iSystems			
Other	Packages	TSSOP-16 QFN-32	QFN-32 LQFP-48	QFN-32 LQFP-64	LQFP-64 LQFP-100	LQFP-64 LQFP-100 MAPBGA-100	MAPBGA-100 LQFP-100* LQFP-144 LQFP-176	MAPBGA-100 LQFP-144 LQFP-176

LEGEND:
- o Not implemented.
- ● Available on the device.
- * Optional.

Figure 2-2. S32K1xx Product Series comparison

Freescale Confidential Proprietary Only

2.6 Module Functional Categories

The modules on this device are grouped into functional categories. The following sections describe the modules assigned to each category in more detail.

Table 2-2. Module functional categories

Module category	Description
ARM® Cortex®-M4F core	• 32-bit MCU core from ARM's Cortex-M class adding DSP instructions and single-precision floating point unit based on ARMv7 architecture
System	• System integration module • Power management and mode controllers • Multiple power modes available based on high speed run, run, wait, stop, and power-down modes • Miscellaneous control module • Crossbar switch • Memory protection unit • Peripheral bridge • Direct memory access (DMA) controller with multiplexer to increase available DMA requests. DMA can now handle transfers in VLPS mode • External watchdog monitor • Watchdog

Memories	• Internal memories include:
	  • Program flash memory
	  • FlexMemory
	  • FlexNVM
	  • FlexRAM
	  • SRAM

Clocks	• Multiple clock generation options available from internally- and externally-generated clocks • System oscillator to provide clock source for the MCU
Security	• Cyclic Redundancy Check (CRC) module for error detection • Internal watchdog (WDOG) with independent clock source • External watchdog monitor (EWM) module • Error-correcting code (ECC) on Flash and SRAM memories • 128-bit unique identification (ID) number • Memory Protection Unit (MPU) module • ADC self test feature
Analog	• High speed analog-to-digital converter (ADC) • Comparator (CMP), containing 8 bit reference DAC • Bandgap voltage reference (1V reference voltage)
Timers	• Programmable delay block (PDB) • FlexTimers • Low-power periodic interrupt timer (LPIT) • Low power timer (LPTMR) • Independent real time clock (RTC)
Communications	• FlexCAN • Low-power Serial peripheral interface (LPSPI) • Low-power Inter-integrated circuit (LPI^{2}C) • Low-power UART (LPUART) • FlexIO
Human-Machine Interfaces (HMI)	• General purpose input/output controller (GPIO)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

2.6.1 ARM® Cortex®-M4F Core Modules

The following core modules are available on this device.

Table 2-3. Core modules

Module	Description
ARM Cortex-M4F	The ARM® Cortex®-M4F is the newest member of the Cortex M Series of processors targeting microcontroller cores focused on very cost sensitive, deterministic, interrupt driven environments. The Cortex M4F processor is based on the ARMv7 Architecture and Thumb®-2 ISA and is upward compatible with the Cortex M3, Cortex M1, and Cortex M0 architectures. Cortex M4F improvements include an ARMv7 Thumb-2 DSP (ported from the ARMv7-A/R profile architectures) providing 32-bit instructions with SIMD (single instruction multiple data) DSP style multiply-accumulates and saturating arithmetic.
Floating point unit (FPU)	A single-precision floating point unit (FPU) that is compliant to the *IEEE Standard for Floating-Point Arithmetic* (IEEE 754).
NVIC	The ARMv7-M exception model and nested-vectored interrupt controller (NVIC) implement a relocatable vector table supporting many external interrupts, a single non-maskable interrupt (NMI), and priority levels. The NVIC replaces shadow registers with equivalent system and simplified programmability. The NVIC contains the address of the function to execute for a particular handler. The address is fetched via the instruction port allowing parallel register stacking and look-up. The first sixteen entries are allocated to ARM internal sources with the others mapping to MCU-defined interrupts.
AWIC	The primary function of the Asynchronous Wake-up Interrupt Controller (AWIC) is to detect asynchronous wake-up events in stop modes and signal to clock control logic to resume system clocking. After clock restart, the NVIC observes the pending interrupt and performs the normal interrupt or event processing.
Debug interfaces	Most of this device's debug is based on the ARM CoreSight™ architecture. Four debug interfaces are supported: • IEEE 1149.1 JTAG • Serial Wire Debug (SWD) • ARM Real-Time Trace Interface

For Ass

2.6.2 System Modules

The following system modules are available on this device.

Table 2-4. System modules

Module	Description
System integration module (SIM)	The SIM includes integration logic and several module configuration settings.
System mode controller (SMC)	The SMC provides control and protection on entry and exit to each power mode, control for the Power management controller (PMC), and reset entry and exit for the complete MCU.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 2-4. System modules (continued)

Module	Description
Power management controller (PMC)	The PMC provides the user with multiple power options that allow the user to optimize power consumption for the level of functionality needed. Includes power-on-reset (POR) and integrated low voltage detect (LVD) with reset (brownout) capability and selectable LVD trip points.
Miscellaneous control module (MCM)	The MCM includes integration logic.
Crossbar switch (AXBS-Lite)	The AXBS-Lite connects bus masters and bus slaves, allowing all bus masters to access different bus slaves simultaneously and providing arbitration among the bus masters when they access the same slave.
Memory protection unit (MPU)	The MPU provides memory protection and task isolation. It concurrently monitors all bus master transactions for the slave connections.
Peripheral bridge (AIPS-Lite)	The AIPS-Lite converts the crossbar switch interface to an interface to access a majority of peripherals on the device.
DMA multiplexer (DMAMUX)	The DMA multiplexer selects from many DMA requests down to a smaller number for the DMA controller.
enhanced Direct Memory Access controller (eDMA)	The eDMA controller provides programmable channels with transfer control descriptors for data movement via dual-address transfers for 8-bit, 16-bit, 32-bit, 16-byte and 32-byte data values.
External watchdog monitor (EWM)	The EWM is a redundant mechanism to the software watchdog module that monitors both internal and external system operation for fail conditions.
Software watchdog (WDOG)	The WDOG monitors internal system operation and forces a reset in case of failure. It can run from an independent 128 kHz low power oscillator with a programmable refresh window to detect deviations in program flow or system frequency.

ssment

2.6.3 Memories and Memory Interfaces

The following memories and memory interfaces are available on this device.

Table 2-5. Memories and memory interfaces

Module	Description

Flash memory	• Program flash memory — non-volatile flash memory that can execute program code
	• FlexMemory — encompasses the following memory types:
	  • FlexNVM — Non-volatile flash memory that can execute program code, store data, or backup EEPROM data
	  • FlexRAM — RAM memory that can be used as traditional SRAM or as high-endurance EEPROM storage, and also accelerates flash programming

Flash memory controller	Manages the interface between the device and the on-chip flash memory.
SRAM	Internal system RAM.
Local memory controller (LMEM)	Manages simultaneous accesses to system RAM by multiple master peripherals and core. Controls cache which improves system performance by providing single-cycle access to the instruction and data pipelines.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

2.6.4 Clocks

The following clock modules are available on this device.

Table 2-6. Clock modules

Module	Description
System clock generator (SCG)	The SCG provides several clock sources for the MCU that include: • Phase-locked loop (PLL) — Voltage-controlled oscillator (VCO) • Fast internal reference clock (FIRC) — An internally generated 48-60 MHz clock, can be used as a clock source for other on-chip peripherals • Slow internal reference clock (SIRC) — An internally generated 8 MHz or 2 MHz clock, can be used as a clock source for other on-chip peripherals • System oscillator (OSC)— The system oscillator, in conjunction with an external crystal or resonator, generates a reference clock for the MCU
Low Power Oscillator (LPO)	An internally generated low power clock with typical frequency of 128 kHz, can be used as clock source for modules working under low power modes.
Peripheral Clock Control (PCC)	Clock selection for most modules is controlled by the PCC module.

2.6.5 Security and Integrity modules

The following security and integrity modules are available on this device:

Table 2-7. Security and integrity modules

Module	Description
Cyclic Redundancy Check (CRC)	Hardware CRC generator circuit using 16/32-bit shift register. Error detection for all single, double, odd, and most multi-bit errors, programmable initial seed value, and optional feature to transpose input data and CRC result via transpose register.

2.6.6 Analog modules

The following analog modules are available on this device:

Table 2-8. Analog modules

Module	Description
12-bit analog-to-digital converters (ADC)	12-bit successive-approximation ADC
Analog comparators (CMP)	Compares two analog input voltages across the full range of the supply voltage.
8-bit digital-to-analog converters (DAC) within CMP	256-tap resistor ladder network which provides a selectable voltage reference for applications where voltage reference is needed.

Review Draft
Freescale Confidential Proprietary - Non-Disclosure Agreement required

2.6.7 Timer modules

The following timer modules are available on this device:

Table 2-9. Timer modules

Module	Description
Programmable delay block (PDB)	• 16-bit resolution • 3-bit prescaler • Positive transition of trigger event signal initiates the counter • Supports multiple triggered delay output signals, each with an independently-controlled delay from the trigger event • Continuous-pulse output or single-shot mode supported, each output is independently enabled, with possible trigger events • Supports bypass mode • Supports DMA
Flexible timer module (FTM)	• Selectable FTM source clock, programmable prescaler • 16-bit counter supporting free-running or initial/final value, and counting is up or up-down • Input capture, output compare, and edge-aligned and center-aligned PWM modes • Operation of FTM channels as pairs with equal outputs, pairs with complementary outputs, or independent channels with independent outputs • Deadtime insertion is available for each complementary pair • Generation of hardware triggers • Software control of PWM outputs • Up to 4 fault inputs for global fault control • Configurable channel polarity • Programmable interrupt on input capture, reference compare, overflowed counter, or detected fault condition and reload opportunity. • Quadrature decoder with input filters, relative position counting, and interrupt on position count or capture of position count on external event • DMA support for FTM events
Low-power periodic interrupt timer (LPIT)	• Four general purpose interrupt timers • Interrupt timers for triggering ADC conversions • 32-bit counter resolution • The counter is clocked by an asynchronous clock that can remain enabled in low power modes. • DMA support • Supports chaining of Timer channels
Low power timer (LPTMR)	• Selectable clock for prescaler/glitch filter of 128 kHz (internal LPO), or internal reference clock • Configurable Glitch Filter or Prescaler with 16-bit counter • 16-bit time or pulse counter with compare • Interrupt generated on Timer Compare • Hardware trigger generated on Timer Compare
Real-time clock (RTC)	• 32-bit seconds counter with 32-bit Alarm • 16-bit Prescaler with compensation that can correct errors between 0.12 ppm and 3906 ppm

Freescale Confidential Proprietary - Non-Disclosure Agreement required

2.6.8 Communication interfaces

The following communication interfaces are available on this device:

Table 2-10. Communication modules

Module	Description
Low-power Serial peripheral interface (LPSPI)	Synchronous serial bus for communication to an external device. LPSPI optionally remains functional in low power modes.
Low-power Inter-integrated circuit (LPI2C)	Allows communication between a number of devices. Also supports the System Management Bus (SMBus) Specification, version 2. LPI2C optioanally remains functionals in low power modes.
Low-power Universal asynchronous receiver/transmitters (LPUART)	Asynchronous serial bus communication interface, supporting LIN master and slave operation. LPUART optionally remains functional in low power modes.
FlexIO	The FlexIO module is capable of supporting a wide range of serial/parallel communication protocols.

2.6.9 Human-machine interfaces

The following human-machine interfaces (HMI) are available on this device:

Table 2-11. HMI modules

Module	Description
General purpose input/output (GPIO)	All general purpose input or output (GPIO) pins are capable of interrupt and DMA request generation.

For Assessment Purpose

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 3
Core Overview

3.1 ARM Cortex-M4F Core Configuration

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

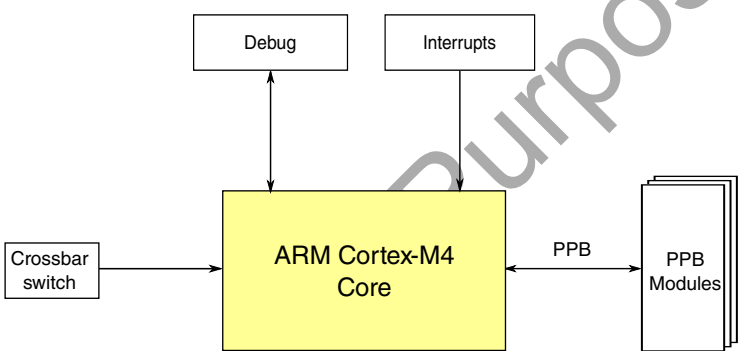


Figure 3-1. Core configuration

Table 3-1. Reference links to related information

Topic	Related module	Reference
Full description	ARM Cortex-M4F core	ARM Cortex-M4F 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	IEEE 1149.1 JTAG Serial Wire Debug (SWD) ARM Real-Time Trace Interface	Debug
Interrupts	Nested Vectored Interrupt Controller (NVIC)	NVIC

Table continues on the next page...

Internal Purposes Only

Table 3-1. Reference links to related information (continued)

Topic	Related module	Reference
Private Peripheral Bus (PPB) module	Miscellaneous Control Module (MCM)	MCM
Private Peripheral Bus (PPB) module	Single-precision floating point unit (FPU)	FPU

3.1.1 Buses, interconnects, and interfaces

The ARM Cortex-M4 core has four buses as described in the following table.

Bus name	Description
Instruction code (ICODE) bus	The ICODE and DCODE buses are muxed. This muxed bus is called the CODE bus and is connected to the crossbar switch via a single master port.
Data code (DCODE) bus	
System bus	The system bus is connected to a separate master port on the crossbar.
Private peripheral (PPB) bus	The PPB provides access to these modules: • ARM modules such as the NVIC, ITM, DWT, FBP, and ROM table • NXP Miscellaneous Control Module (MCM)

3.1.2 System Tick Timer

The System Tick Timer's clock source is always the core clock, CORE_CLK. This results in the following:

- The CLKSOURCE bit in SysTick Control and Status register is always set to select the core clock.
- Because the timing reference (CORE_CLK) is a variable frequency, the TENMS bit in the SysTick Calibration Value Register is always zero.
- The NOREF bit in SysTick Calibration Value Register is always set, implying that CORE_CLK is the only available source of reference timing.

3.1.3 Debug facilities

This device has extensive debug capabilities including run control and tracing capabilities. The standard ARM debug port that supports JTAG and SWD interfaces.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

3.1.4 Caches

This device includes one 4 KB code cache to minimize the performance impact of memory access latencies. The code cache exists on I/D bus, and there is no cache on the system bus.

Features of the cache are:

- 2-way set associative
- 4 word lines
- Lines can be individually flushed
- Entire cache can be flushed at once

3.1.4.1 Control

For control purposes the cache can be in one of these states:

1. Write Back / Write Allocate (WBWA)
2. Write Through
3. No cache

For each defined region there will be 2 bits allocated on the control register that determines the cache state for the memory region associated with this section. The user can only "lower" the cache attribute, given the fixed relationship of WBWA > WT > NC - so, you can demote a WBWA region to either WT or NC, you can demote a WT space to NC. In order to change the state upwards a system reset is required.

3.1.5 Core privilege levels

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

Table 3-2. Terms used

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

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

3.2 Nested Vectored Interrupt Controller (NVIC) Configuration

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

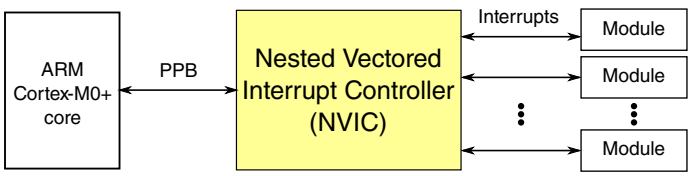


Figure 3-2. NVIC configuration

Table 3-3. Reference links to related information

Topic	Related module	Reference
Full description	Nested Vectored Interrupt Controller (NVIC)	ARM Cortex-M Technical Reference Manual
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Private Peripheral Bus (PPB)	ARM Cortex-M4 core	ARM Cortex-M4 core

3.2.1 Interrupt priority levels

This device supports 16 priority levels for interrupts. Therefore, in the NVIC each source in the IPR registers contains 4 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	IRQ2				0	0	0	0	IRQ1				0	0	0	0	IRQ0				0	0	0	0
W																																

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 for the $\overline{\text{NMI}}$ function to generate the non-maskable interrupt request.

For Assessment Purposes Only

Refer to S32K144_IO_Signal_Description_Input_Multiplexing.xlsx attached with the Reference Manual, for details on NMI pad.

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

Suppose you need to configure the low-power timer (LPTMR) interrupt. The following table is an excerpt of the LPTMR row from the S32K1xx_DMA_INT_mapping.xlsm attached with the Reference Manual for details.

Table 3-5. LPTMR interrupt vector assignment

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0128	74	58	1	14	Low Power Timer	—

1. Indicates the NVIC's interrupt source number.
2. Indicates the NVIC's ISER, ICER, ISPR, ICPR, and IABR register number used for this IRQ. The equation to calculate this value is: IRQ div 32
3. 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:
  - NVICISER1
  - NVICICER1
  - NVICISPR1
  - NVICICPR1
  - NVICIABR1
  - NVICIPR14
- To determine the particular IRQ's bitfield location within these particular registers:
  - NVICISER1, NVICICER1, NVICISPR1, NVICICPR1, NVICIABR1 bit location = IRQ mod 32 = 26
  - NVICIPR14 bitfield starting location = 8 × (IRQ mod 4) + 4 = 20

    Since the NVICIPR bitfields are 4-bit wide (16 priority levels), the NVICIPR14 bitfield range is 20-23

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

- NVICISER1[26]
- NVICICER1[26]
- NVICISPR1[26]
- NVICICPR1[26]

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- NVICIABR1[26]
- NVICIPR14[23:20]

3.3 Asynchronous Wake-up Interrupt Controller (AWIC) Configuration

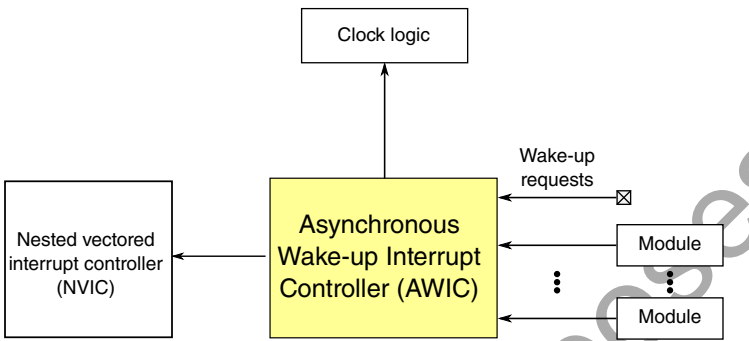


Figure 3-3. Asynchronous Wake-up Interrupt Controller configuration

Table 3-6. Reference links to related information

Topic	Related module	Reference
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
	Nested Vectored Interrupt Controller (NVIC)	NVIC
Wake-up requests		AWIC wake-up sources

3.3.1 Wake-up sources

Table 3-7. AWIC Stop and VLPS Wake-up Sources

Wake-up source	Description
Available system resets	RESET pin, WDOG, JTAG
Pin interrupts	Port Control Module - Any enabled pin interrupt is capable of waking the system
ADCx	ADCx is optional functional with clock source from SIRC or OSC
CMP	Functional in Stop/VLPS modes with clock source from SIRC or OSC
LPI2C	Functional in Stop/VLPS modes with clock source from SIRC or OSC
LPUART	Functional in Stop/VLPS modes with clock source from SIRC or OSC
LPSPI	Functional in Stop/VLPS modes with clock source from SIRC or OSC

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 3-7. AWIC Stop and VLPS Wake-up Sources (continued)

Wake-up source	Description
LPTMR	Functional in Stop/VLPS modes
RTC	Functional in Stop/VLPS modes
CAN	CAN stop wakeup
NMI	Non-maskable interrupt

3.4 FPU Configuration

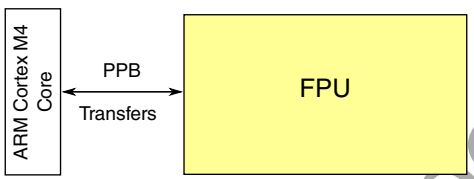


Figure 3-4. FPU configuration

Table 3-8. Reference links to related information

Topic	Related module	Reference
Full description	FPU	ARM Cortex-M4 Technical Reference Manual
System memory map		System memory map
Clocking		Clock Distribution
Power Management		Power Management
Transfers Private Peripheral Bus (PPB)	ARM Cortex M4 core	ARM Cortex-M4 core

3.5 JTAG Controller Configuration

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

For Asses
Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

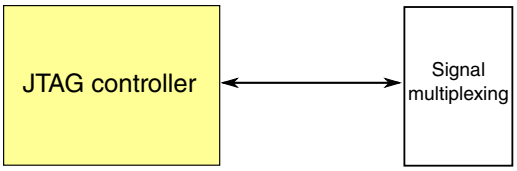


Figure 3-5. JTAG controller configuration

Table 3-9. Reference links to related information

Topic	Related module	Reference
Full description	JTAGC	JTAGC
Signal multiplexing	Port control	Refer to the S32K144_IO_Signal_Description_Input_Multiplexing.xlsx attached to Reference Manual for details.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 4
Memories and Memory Interfaces

4.1 Flash Memory Configuration

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

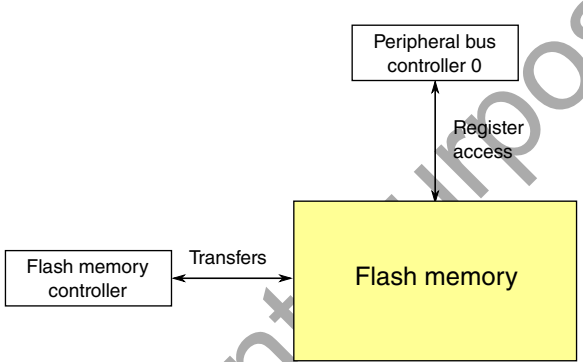


Figure 4-1. Flash memory configuration

Table 4-1. Reference links to related information

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

4.1.1 Flash memory types

This device contains the following types of flash memory:

For Development Purposes Only

- Program flash memory — non-volatile flash memory that can execute program code
- FlexMemory — encompasses the following memory types:
  - FlexNVM — Non-volatile flash memory that can execute program code, store data, or backup EEPROM data
  - FlexRAM — RAM memory that can be used as SRAM or as high-endurance EEPROM storage, and also accelerates flash programming

4.1.2 Flash Memory Sizes

The devices covered in this document contain:

- 512KB of program flash consisting of 4 KB sectors
- 1 block (64 KB) of FlexNVM consisting of 2 KB sectors
- 1 block (4 KB) of FlexRAM (can also be used as System RAM)

4.1.3 Flash Memory Map

The various flash memories 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.

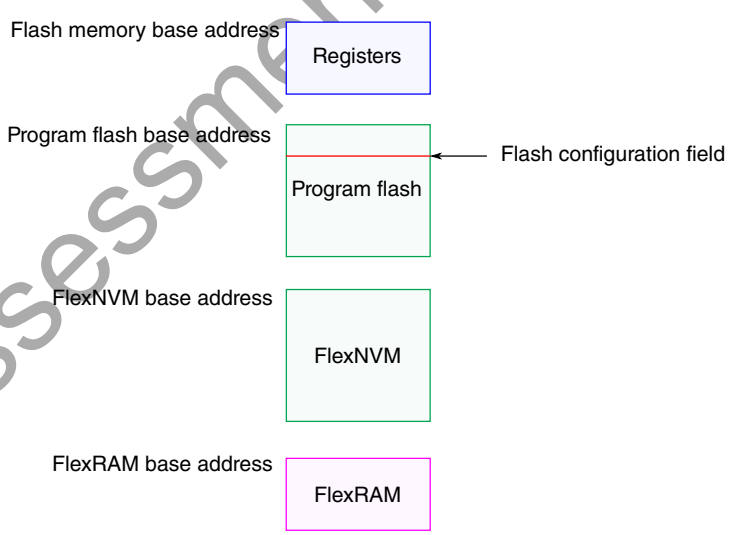


Figure 4-2. Flash memory map

4.1.4 Flash Security

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

For Assessment Purposes Only

4.1.5 Flash Program Restrictions

The flash memory on this device should not be programmed or erased while operating in VLPR power mode.

4.1.6 Flash Modes

The flash memory is always configured in NVM normal. There are no operating conditions in which the flash is configured for NVM special mode.

4.1.7 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 the entire flash memory may be erased external to the flash memory from the SWJ-DP debug port by setting DAP_CONTROL[0]. DAP_STATUS[0] is set to indicate the mass erase command has been accepted. DAP_STATUS[0] is cleared when the mass erase completes.

4.1.8 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 for details of its definition.

4.2 Flash Memory Controller Configuration

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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

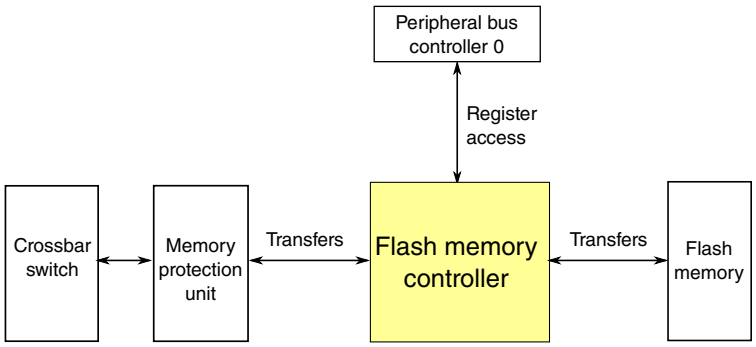


Figure 4-3. Flash memory controller configuration

Table 4-2. Reference links to related information

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

4.2.1 Number of masters

The Flash Memory Controller supports up to eight crossbar switch masters. However, this device has a different number of crossbar switch masters. See Crossbar Switch Master Assignments for details on the master port assignments.

4.3 SRAM Configuration

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

Table 4-3. Reference links to related information

Topic	Related module	Reference
Full description	SRAM	SRAM
System memory map		System memory map
Clocking	System Clock Generator	Clock Distribution

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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

Topic	Related module	Reference
Transfers	SRAM controller	SRAM controller
	ARM Cortex-M4 core	ARM Cortex-M4 core
	Memory protection unit	Memory protection unit

4.3.1 SRAM sizes

This device contains SRAM tightly coupled to the ARM Cortex-M4F core. The on-chip SRAM is split into SRAM_L and SRAM_U regions where the SRAM_L and SRAM_U ranges form a contiguous block in the memory map anchored at address 0x2000_0000. As such:

- SRAM_L is anchored to 0x1FFF_FFFF and occupies the space before this ending address.
- SRAM_U is anchored to 0x2000_0000 and occupies the space after this beginning address.

NOTE

Misaligned accesses across the 0x2000_0000 boundary are not supported in the ARM Cortex-M4F architecture.

The amount of SRAM for the devices covered in this document is shown in the following table.

Device	SRAM_L size (KB)	SRAM_U size (KB)[1]	Total SRAM (KB)[1]	FlexRAM (KB)[1]
S32K148	256	128	124	4
	192	96	92	4
S32K146	128	64	60	4
	96	48	44	4
S32K144	64	32	28	4
	48	28	16	4
S32K142	32	16	12	4
	16	8	4	4

1. Refer to S32K1xx_memory_map.xlsx for specific address locations

NOTE

No ECC and access error will be generated for 4 KB FlexRAM used as System RAM . The region 0x2000_6FFF -

Assessment Purposes Only

0x2000_7FFF is reserved however no access error is generated for this region

4.3.2 SRAM retention in low power modes

The SRAM is retained power on to all power modes on this device.

SRAM contents can be retained across functional resets by using SRAML_RETEN/ SRAMU_RETEN of SIM CHIPCTL register . Following steps need to be ensured for that:

1. Application code needs to configure RCM (RCM_SRIE – System Reset Interrupt Enable Register) to delay the reset and instead generate an interrupt at every reset request.
2. On this interrupt, application code must configure itself into RUN mode and then needs to configure these bits. This logic will ensure that any access to SRAM is blocked after these bits are written and there is no way for memory to be corrupted during reset assertion.
3. Software must continue to stay in run mode till reset asserts.
4. After coming out of reset , application code should read these bits and write “1” to them to allow accesses to SRAM.

NOTE

To use this feature software initialization should ensure that no accesses to RAM that is retained are made until the SRAML_RETEN/SRAMU_RETEN bits are 1.

4.3.3 SRAM accesses

The SRAM is split into two logical arrays that are 32-bits wide.

- SRAM_L — Accessible by the code bus of the Cortex-M4F core and by the backdoor port.
- SRAM_U — Accessible by the system bus of the Cortex-M4F core and by the backdoor port.

The backdoor port makes the SRAM accessible to the non-core bus masters (such as DMA).

The following figure illustrates the SRAM accesses within the device.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

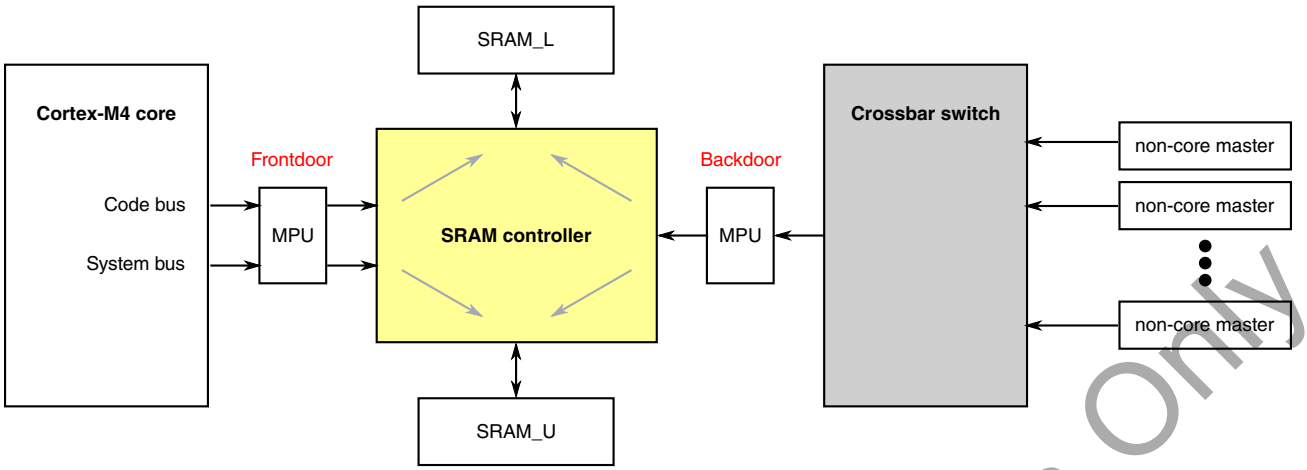


Figure 4-4. SRAM access diagram

The following simultaneous accesses can be made to different logical regions of the SRAM:

- Core code and core system
- Core code and non-core master
- Core system and non-core master

Core code	Core system	Non-core master
SRAM_L	SRAM_U	-
SRAM_L	-	SRAM_U
-	SRAM_U	SRAM_L

NOTE

Burst-access cannot occur across the 0x2000_0000 boundary that separates the two SRAM arrays. The two arrays should be treated as separate memory ranges for burst accesses.

4.3.4 SRAM arbitration and priority control

The MCM_CPCR register controls the arbitration and priority schemes for the two SRAM arrays.

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 5
Memory Map

5.1 Introduction

This device contains various memories and memory-mapped peripherals which are located in one 32-bit contiguous memory space. This chapter describes the memory and peripheral locations within that memory space.

5.2 Aliased bit-band regions

The SRAM_U, AIPS-Lite, and general purpose input/output (GPIO) module resources reside in the Cortex-M4F processor bit-band regions. [1]

The processor also includes two 32 MB aliased bit-band regions associated with the two 1 MB bit-band spaces. Each 32-bit location in the 32 MB space maps to an individual bit in the bit-band region. A 32-bit write in the alias region has the same effect as a read-modify-write operation on the targeted bit in the bit-band region.

Bit 0 of the value written to the alias region determines what value is written to the target bit:

- Writing a value with bit 0 set writes a 1 to the target bit.
- Writing a value with bit 0 clear writes a 0 to the target bit.

A 32-bit read in the alias region returns either:

- a value of 0x0000_0000 to indicate the target bit is clear
- a value of 0x0000_0001 to indicate the target bit is set

1. The S32K series and the software drivers support bit-banding, but ARM no longer promote its usage hence it is not recommended to use bit-banding.

For Assessment Purposes Only

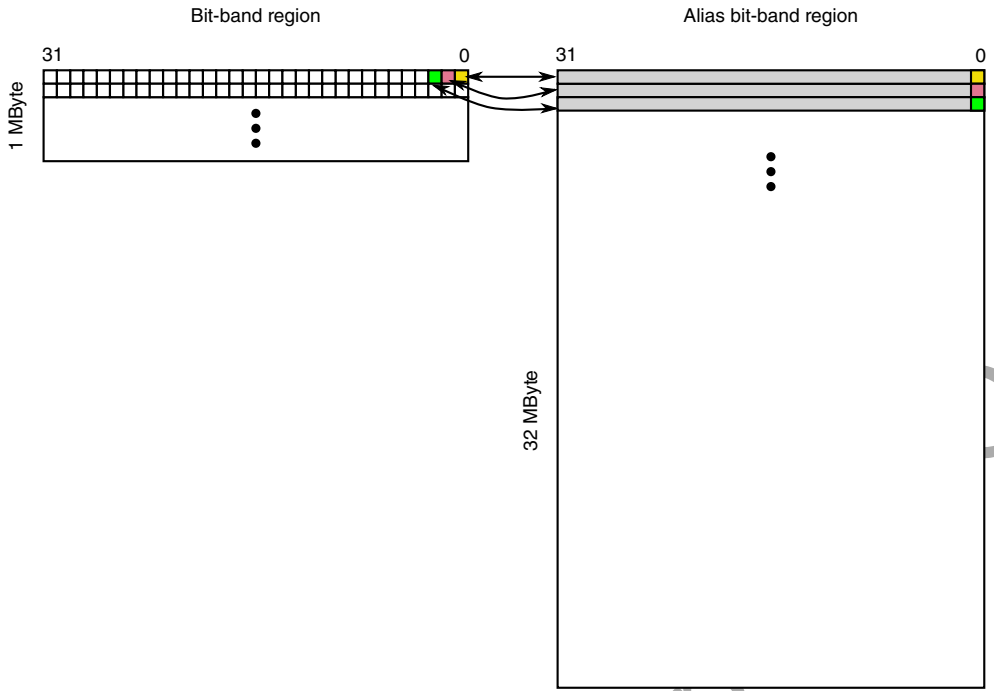


Figure 5-1. Alias bit-band mapping

NOTE

Each bit in bit-band region has an equivalent bit that can be manipulated through bit 0 in a corresponding long word in the alias bit-band region.

5.3 Flash Memory Map

The various flash memories 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.

For Assessment Purp Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

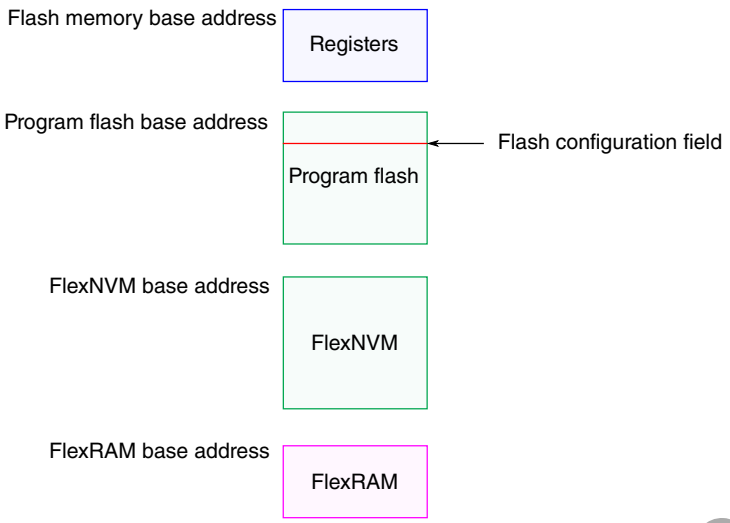


Figure 5-2. Flash memory map

5.3.1 Alternate Non-Volatile IRC User Trim Description

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

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

5.4 SRAM memory map

The on-chip RAM is split in two regions: SRAM_L and SRAM_U. The RAM is implemented such that the SRAM_L and SRAM_U ranges form a contiguous block in the memory map. See S32K1xx_memory_map.xlsx attached to Reference Manual 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.

For Assessment Purposes Only

5.5 Peripheral bridge (AIPS-Lite) memory map

The peripheral memory map is accessible via a crossbar slave port.

There are three regions assosiated with peripheral space:

- 0x4000_0000–0x4001_FFFF: 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.
- 0x4002_0000–0x4007_FFFF: 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.
- 0x400F_F000: a 4 KB region for accessing the GPIO module. This block is connected to the AMBA bus via the port splitter and provides direct master access without incurring wait states associated with accesses via the AIPS controllers. The GPIO is implemented only in the upper space of this region (4 kbytes beginning at 0x400F_F000).

Modules that are disabled via their clock gate control bits in the PCC/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.

5.5.1 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

NOTE

One factor contributing to these situations is processor write buffering. The processor architecture has a programmable configuration bit to disable write buffering: ACTLR[DISDEFWBUF]. However, disabling buffered writes is likely to degrade system performance much more than simply performing the required memory serialization for the situations that truly require it.

5.6 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 5-1. PPB memory map

System 32-bit Address Range	Resource
0xE000_0000–0xE000_0FFF	Instrumentation Trace Macrocell (ITM)
0xE000_1000–0xE000_1FFF	Data Watchpoint and Trace (DWT)
0xE000_2000–0xE000_2FFF	Flash Patch and Breakpoint (FPB)
0xE000_3000–0xE000_DFFF	Reserved
0xE000_E000–0xE000_EFFF	System Control Space (SCS) (for NVIC and FPU
0xE000_F000–0xE003_FFFF	Reserved
0xE004_0000–0xE004_0FFF	Trace Port Interface Unit (TPIU)
0xE004_1000–0xE004_1FFF	Reserved
0xE004_2000–0xE004_2FFF	Reserved
0xE004_3000–0xE004_3FFF	Reserved
0xE004_4000–0xE007_FFFF	Reserved
0xE008_0000–0xE008_0FFF	Miscellaneous Control Module (MCM)
0xE008_1000–0xE008_1FFF	Reserved
0xE008_2000–0xE008_2FFF	Cache Controller (LMEM)
0xE008_3000–0xE00F_EFFF	Reserved
0xE00F_F000–0xE00F_FFFF	ARM Core ROM Table[1] - allows auto-detection of debug components

1. The ARM Core ROM table is optionally required by ARM CoreSight debug infrastructure to discover the components on the chip. This ROM table has no any relationship with the MCU Boot ROM.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 6
Clock Distribution

6.1 Introduction

This chapter presents the clock architecture overview of this device, clock distribution, module clocks, and clock terminology. The System Clock Generator (SCG) module is used to generate most of the clocks used by the device. The SCG module controls which clock source (internal references, external crystals, external clocks) is used to derive system clocks. The SCG also divides the selected clock source into a variety of clock domains, including clocks for system bus masters, system bus slaves, and flash memory.

The clock generation circuitry provides several clock dividers and selectors allowing different modules to be clocked at a frequency specific for that module. Clock generation logic also implements module specific clock gating allowing modules to be individually disabled. Thus, allowing optimization for performance or low power.

Various modules have specific clocks that can be generated from FIRC_CLK, SIRC_CLK, SOSC_CLK, SPLL_CLK, or Power Management Controller (PMC) clock signal (LPO128K_CLK). In addition, modules specific clocks that can be configured from alternate sources. Clock selection for most modules is controlled by the PCC module.

6.2 High level clocking diagram

The following diagram shows the high-level clocking architecture and various clock sources for this device.

For Assessment Purposes Only

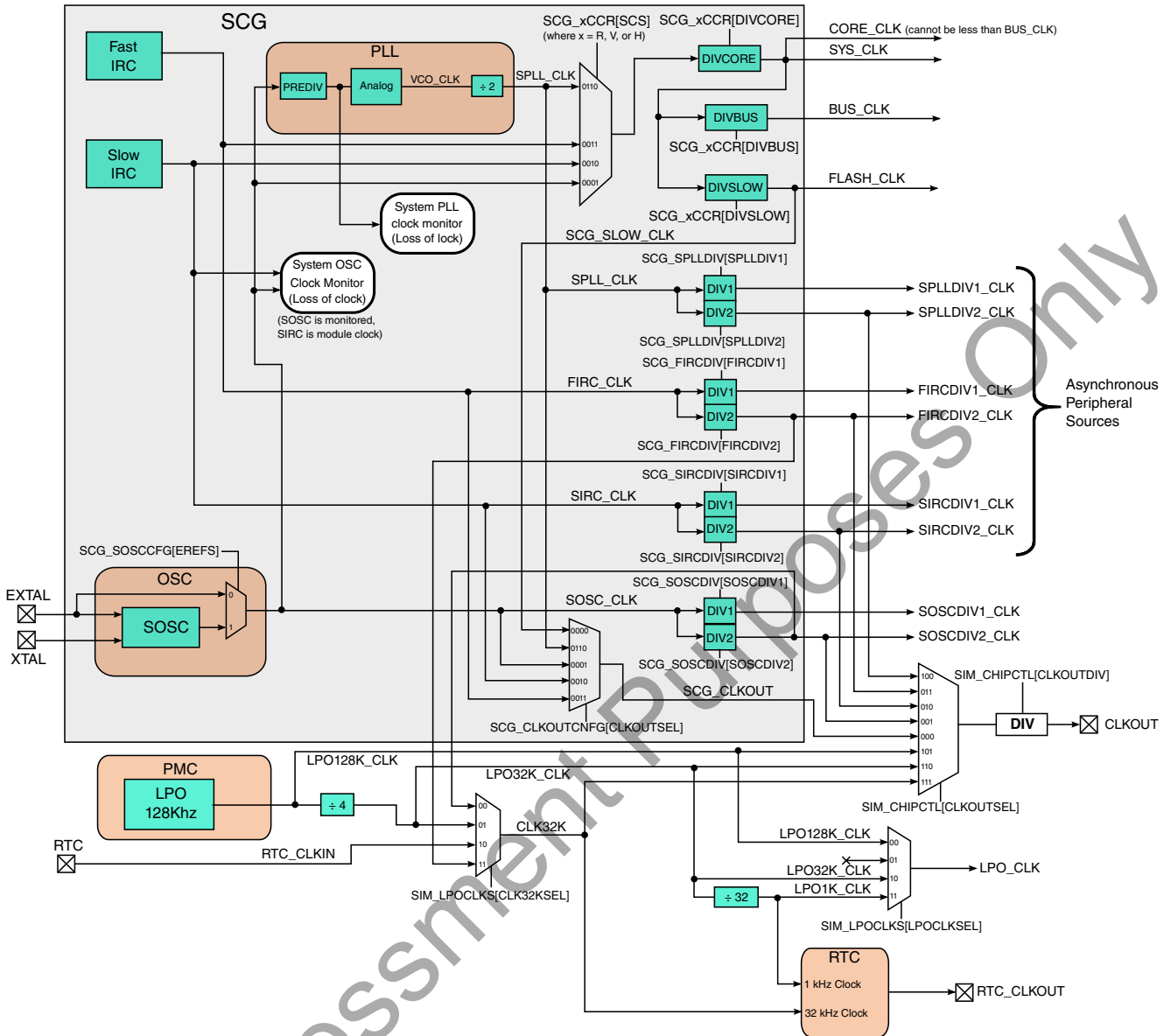


Figure 6-1. Clocking diagram

For Assessment Purposes Only

6.3 Clock definitions

The following table describes clocks shown in Figure 6-1 and other sections of this document.

Table 6-1. Clock descriptions

Clock name	Related clock selector	Related clock divider	Description
CORE_CLK[1]	SCG_xCCR[SCS]	SCG_xCCR[DIVCORE] (÷ 1...16)	Clocks the ARM core, divided by DIVCORE bits inside SCG.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 6-1. Clock descriptions (continued)

Clock name	Related clock selector	Related clock divider	Description
SYS_CLK	SCG_xCCR[SCS]	SCG_xCCR[DIVCORE] (÷ 1...16)	Clocks the Crossbar, NVIC, Flash controller, FTM, PDB, and so on.
BUS_CLK	SCG_xCCR[SCS]	SCG_xCCR[DIVBUS] (÷ 1...16)	Clocks the chip peripherals.
FLASH_CLK (SCG_SLOW_CLK in SCG)	SCG_xCCR[SCS]	SCG_xCCR[DIVSLOW] (÷ 1...8)	Clocks the flash module.
SPLL_CLK	—	—	Output of PLL (VCO_CLK ÷ 2)
SIRC_CLK	—	—	Output clock of Slow IRC.
FIRC_CLK	—	—	Output clock of Fast IRC.
SOSC_CLK	SCG_SOSCCFG[EREFS]	—	System oscillator clock. Can be either the EXTAL pin or output of oscillator (SOSC). **NOTE:** ERCLK/OSCERCLK stands for the same clock source, in some module chapters.
RTC_CLKOUT	—	—	RTC oscillator output driving external pin.
LPO32K_CLK	SIM_LPOCLKS[CLK32SEL]	A fixed divide by 4 of LPO_CLK drives 01b input of CLK32K multiplexer.	Source clock for RTC.
LPO128K_CLK	—	—	Always on low power oscillator clock generated by PMC.
SCG_CLKOUT	SCG_CLKOUTCNFG[CLKOUTSEL]	—	SCG ouptut clock that can be driven by SOSC_CLK, SIRC_CLK, FIRC_CLK, SPLL_CLK, or SCG_SLOW_CLK (FLASH_CLK).
LPO_CLK	SIM_LPOCLKS[LPOCLKSEL]	—	Clock output generated from one of three LPO clocks sources (LPO128K_CLK, LPO32K_CLK, LPO1K_CLK).
CLKOUT	SIM_CHIPCTL[CLKOUTSEL]	SIM_CHIPCTL[CLKOUTDIV] (÷ 1...8)	Selected from one of eight internal clock sources.
VCO_CLK	—	—	VCO output clock within PLL. Its frequency = SPLL_CLK × 2.
SPLLDIV1_CLK	—	SCG_SPLLDIV[SPLLDIV1] (÷ 1, 2, 4, 8, 16, 32, 64, or output disabled)	Divided SPLL_CLK
SPLLDIV2_CLK	—	SCG_SPLLDIV[SPLLDIV2] (÷ 1, 2, 4, 8, 16, 32, 64, or output disabled)	Divided SPLL_CLK
FIRCDIV1_CLK	—	SCG_FIRCDIV[FIRCDIV1] (÷ 1, 2, 4, 8, 16, 32, 64, or output disabled)	Divided FIRC_CLK
FIRCDIV2_CLK	—	SCG_FIRCDIV[FIRCDIV2] (÷ 1, 2, 4, 8, 16, 32, 64, or output disabled)	Divided FIRC_CLK

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 6-1. Clock descriptions (continued)

Clock name	Related clock selector	Related clock divider	Description
SIRCDIV1_CLK	—	SCG_SIRCDIV[SIRCDIV1] (÷ 1, 2, 4, 8, 16, 32, 64, or output disabled)	Divided SIRC_CLK
SIRCDIV2_CLK	—	SCG_SIRCDIV[SIRCDIV2] (÷ 1, 2, 4, 8, 16, 32, 64, or output disabled)	Divided SIRC_CLK
SOSCDIV1_CLK	—	SCG_SPLLDIV[SOSCDIV1] (÷ 1, 2, 4, 8, 16, 32, 64, or output disabled)	Divided SOSC_CLK
SOSCDIV2_CLK	—	SCG_SPLLDIV[SOSCDIV2] (÷ 1, 2, 4, 8, 16, 32, 64, or output disabled)	Divided SOSC_CLK

1. Must not be programmed to less than BUS_CLK.

6.4 Internal clocking requirements

The clock dividers are programmed via the SCG module's clock divider registers. The following requirements must be met when configuring the clocks for this chip:

- CORE_CLK and SYS_CLK clock frequency must be 112 MHz or less in HSRUN mode and 80 MHz in normal RUN mode (but not configured to be less than BUS_CLK).
- BUS_CLK frequency must be programmed to 56 MHz or less in HSRUN, 48 MHz or less in RUN, and an integer divide of the CORE_CLK.
- FLASH_CLK frequency must be programmed to 28 MHz or less in HSRUN, 26.67 MHz or less in RUN to BUS_CLK, and an integer divide of the CORE_CLK. The core clock to flash clock ratio is limited to a max value of 8.

The following are a few of the common clock configurations for this chip in the four clocking modes:

Option 1: Slow RUN (typically using the undivided FIRC)[1], using the following Memory Map/Register Definition register settings:

- SCG_RCCR[SCS] = 0011b
- SCG_RCCR[DIVCORE] = 0000b
- SCG_RCCR[DIVBUS] = 0000b
- SCG_RCCR[DIVSLOW] = 0001b

1. Default configuration after reset. FIRC_CLK = 48 MHz.

For Assessment Purpos

Table 6-2. Slow RUN example

Clock	Frequency
CORE_CLK	48 MHz
SYS_CLK	48 MHz
BUS_CLK	48 MHz (max freq. in RUN mode)
FLASH_CLK	24 MHz

Option 2: Slow RUN (with FIRC_CLK = 60 MHz), using the following Memory Map/Register Definition register settings:

- SCG_RCCR[SCS] = 0011b
- SCG_RCCR[DIVCORE] = 0000b
- SCG_RCCR[DIVBUS] = 0001b
- SCG_RCCR[DIVSLOW] = 0010b

Table 6-3. Slow RUN example

Clock	Frequency
CORE_CLK	60 MHz
SYS_CLK	60 MHz
BUS_CLK	30 MHz
FLASH_CLK	20 MHz

Option 3: Normal RUN (with VCO_CLK = 320 MHz, SPLL_CLK = 160 MHz), using the following Memory Map/Register Definition register settings:

- SCG_RCCR[SCS] = 0110b
- SCG_RCCR[DIVCORE] = 0001b
- SCG_RCCR[DIVBUS] = 0001b
- SCG_RCCR[DIVSLOW] = 0010b

Table 6-4. Normal RUN example

Clock	Frequency
CORE_CLK	80 MHz
SYS_CLK	80 MHz
BUS_CLK	40 MHz
FLASH_CLK	26.67 MHz (max freq. in RUN mode)

Option 4: High Speed RUN (with VCO_CLK = 224 MHz, SPLL_CLK = 112 MHz), using the following Memory Map/Register Definition register settings:

- SCG_HCCR[SCS] = 0110b

For Assessment Purposes Only

- SCG_HCCR[DIVCORE] = 0000b
- SCG_HCCR[DIVBUS] = 0001b
- SCG_HCCR[DIVSLOW] = 0011b

Table 6-5. High Speed RUN example

Clock	Frequency
CORE_CLK	112 MHz
SYS_CLK	112 MHz
BUS_CLK	56 MHz
FLASH_CLK	28 MHz

NOTE

All frequencies listed in table above are maximum for HSRUN mode.

Option 5: Very Low Power RUN, VLPR (with SIRC_CLK or SOSC_CLK = 4 MHz), using the following Memory Map/Register Definition register settings:

- SCG_VCCR[SCS] = 0010b (SIRC_CLK), or 0001b (SOSC_CLK)
- SCG_VCCR[DIVCORE] = 0000b
- SCG_VCCR[DIVBUS] = 0000b
- SCG_VCCR[DIVSLOW] = 0011b

Table 6-6. Very low power RUN example

Clock	Frequency
CORE_CLK	4 MHz
SYS_CLK	4 MHz
BUS_CLK	4 MHz
FLASH_CLK	1 MHz

NOTE

All frequencies listed in table above are maximum for VLPR mode.

NOTE

All asynchronous clock sources will also be restricted to 4 MHz as configured in SCG_SPLLDIV, SCG_FIRCDIV, SCG_SOSCDIV, and SCG_SIRCDIV.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

6.4.1 Clock divider values after reset

The default configuration out of reset has the CPU clocked by the Fast IRC (FIRC_CLK). The clocks (for example, CORE_CLK, FLASH_CLK, and BUS_CLK) are configured in the SCG module (see Memory Map/Register Definition).

6.4.2 HSRUN mode clocking

Clock dividers should not be modified while the chip is operating in HSRUN mode. They must be configured prior to entering HSRUN mode to guarantee:

- CORE_CLK/SYS_CLK is less than or equal to 112MHz
- BUS_CLK is less than or equal to 56 MHz
- FLASH_CLK is less than or equal to 28 MHz

6.4.3 VLPR mode clocking

Clock dividers should not be modified while the chip is operating in VLPR mode. They must be configured prior to entering VLPR mode to guarantee:

- CORE_CLK/SYS_CLK and BUS_CLK are less than or equal to 4 MHz
- FLASH_CLK is less than or equal to 1 MHz

NOTE

All asynchronous clock sources will also be restricted to 4 MHz as configured in SCG_SPLLDIV, SCG_FIRCDIV, SCG_SOSCDIV, and SCG_SIRCDIV.

6.5 Clock Gating

The clock to each module can be individually gated on and off using the PCC module. After any reset, PCC disables the clock to the corresponding module to conserve power. Prior to initializing a module, set the corresponding clock gating control bits in PCC 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.

For Assessment Purposes Only

6.6 Module clocks

The following table summarizes the clocks that can be used by each of the modules.

Table 6-7. Peripheral clock summary

Module name	Bus interface clock	Bus interface clock cating	Peripheral functional clock	Additonal clocks	Comments and maximum frequencies
		Gated by [CGC] of PCC	Clocks controlled by [PCS] of PCC		
Communications					
LPUART[0:2]	BUS_CLK	Yes	SPLLDIV2_CLK, FIRCDIV2_CLK, SIRCDIV2_CLK, SOSCDIV2_CLK	—	Maximum frequency governed by BUS_CLK
LPSPI[0:2]	BUS_CLK	Yes	SPLLDIV2_CLK, FIRCDIV2_CLK, SIRCDIV2_CLK, SOSCDIV2_CLK	—	Maximum frequency governed by BUS_CLK
LPI2C0	BUS_CLK	Yes	SPLLDIV2_CLK, FIRCDIV2_CLK, SIRCDIV2_CLK, SOSCDIV2_CLK	—	Maximum frequency governed by BUS_CLK
FlexIO	BUS_CLK	Yes	SPLLDIV2_CLK, FIRCDIV2_CLK, SIRCDIV2_CLK, SOSCDIV2_CLK	—	Maximum frequency governed by BUS_CLK
FlexCAN[0:2]	SYS_CLK	Yes	—	BUS_CLK, SOSCDIV2_CLK	Support 40 MHz from OSC; SYS_CLK must be >1.5x the protocol clock; while synchronous operation (when protocol clock is selected to BUS_CLK) can be done at 1:1 clock frequency.
Timers					
LPTMR	BUS_CLK	Yes	SPLLDIV2_CLK, FIRCDIV2_CLK, SIRCDIV2_CLK, SOSCDIV2_CLK	CLK32K[1], SIRCDIV2_CLK, LPO1K_CLK	Maximum frequency governed by BUS_CLK
LPIT	BUS_CLK	Yes	SPLLDIV2_CLK, FIRCDIV2_CLK, SIRCDIV2_CLK, SOSCDIV2_CLK	—	Maximum frequency governed by BUS_CLK
RTC	BUS_CLK	Yes	—	CLK32K[1], LPO1K_CLK	—
PDB[0:1]	SYS_CLK	Yes	—	—	—

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 6-7. Peripheral clock summary (continued)

Module name	Bus interface clock	Bus interface clock cating	Peripheral functional clock	Additonal clocks	Comments and maximum frequencies
		Gated by [CGC] of PCC	Clocks controlled by [PCS] of PCC		
FlexTimer[0:3]	SYS_CLK	Yes	SPLLDIV1_CLK, FIRCDIV1_CLK, SIRCDIV1_CLK, SOSCDIV1_CLK	RTC_CLK, SYS_CLK, TCLKx	—
System Modules					
Watchdog	BUS_CLK	Yes	—	RTC_CLK, SOSC_CLK, SIRC_CLK	Maximum frequency governed by BUS_CLK
EWM	BUS_CLK	Yes	—	LPO_CLK	—
PMC	BUS_CLK	No	—	LPO_CLK	—
SIM	BUS_CLK	Yes	BUS_CLK	BUS_CLK	—
RCM	BUS_CLK	No	BUS_CLK	BUS_CLK	—
PORT	BUS_CLK	Yes	—	BUS_CLK, LPO_CLK	—
CRC	BUS_CLK	Yes	BUS_CLK	BUS_CLK	—
RGPIO	BUS_CLK	Yes	LPO_CLK	BUS_CLK	—
DMA	SYS_CLK	Yes	SYS_CLK	SYS_CLK	—
Memory Modules					
FTFC	BUS_CLK	Yes	FLASH_CLK	BUS_CLK	—
SYS RAM	SYS_CLK	No	SYS_CLK	SYS_CLK	—
Analog Modules					
ADC[0:1]	BUS_CLK	Yes	SPLLDIV2_CLK, FIRCDIV2_CLK, SIRCDIV2_CLK, SOSCDIV2_CLK	—	50 MHz[2]
ACMP0	BUS_CLK	Yes	BUS_CLK	BUS_CLK	—

1. CLK32K is the output from the multiplexer that uses SIM_LPOCLKS[CLK32KSEL] for souce selection (See Figure 1 for details).
2. 50 MHz is the maximum frequency for the ADC, but FIRC can be configured for 64 MHz operation.

NOTE

The above clock selections are controlled in the PCC module (column 4, 'Peripheral functional clock').

For Asse

6.6.1 PCC connections

The following table summarizes the clocks that can be used by each of the modules.

Table 6-8. Clocking from PCC module

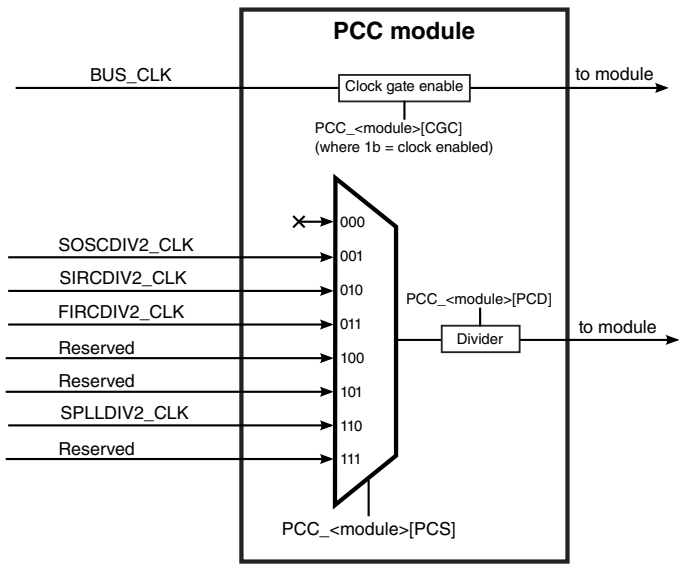
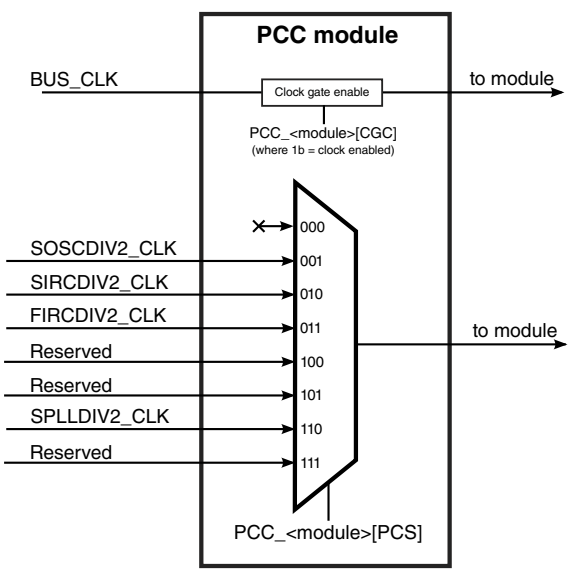
Modules	PCC multiplexer
LPTMR0	
LPSPI[0:2] LPIT FlexIO LPI2C0 LPUART[0:2]	

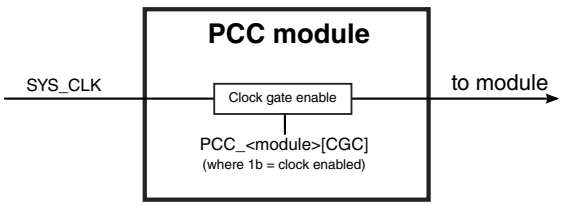
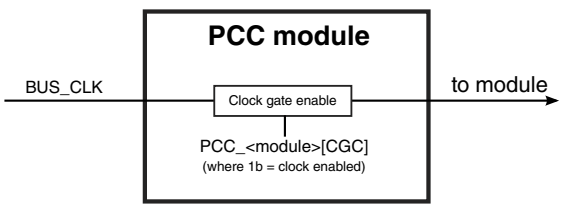
Figure 6-2. PCC connection, BUS_CLK, mux, divider

Figure 6-3. PCC connection with BUS_CLK

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 6-8. Clocking from PCC module (continued)

Modules	PCC multiplexer
PDB[0:1]	 **Figure 6-4. PCC connection, SYS_CLK , no mux, no divider**
DMAMUX0 CRC RGPIO[0:4] (PORT[A:E])	 **Figure 6-5. PCC connection, BUS_CLK, no mux, no divider**

1. The divider also has fractional capabilities, see PCC module chapter for details.

6.6.2 EWM clocks

This table shows the EWM clocks and the corresponding chip clocks.

Table 6-9. EWM clock connections

Module clock	Chip clock
Low Power Clock	128 kHz LPO Clock (LPO_CLK)

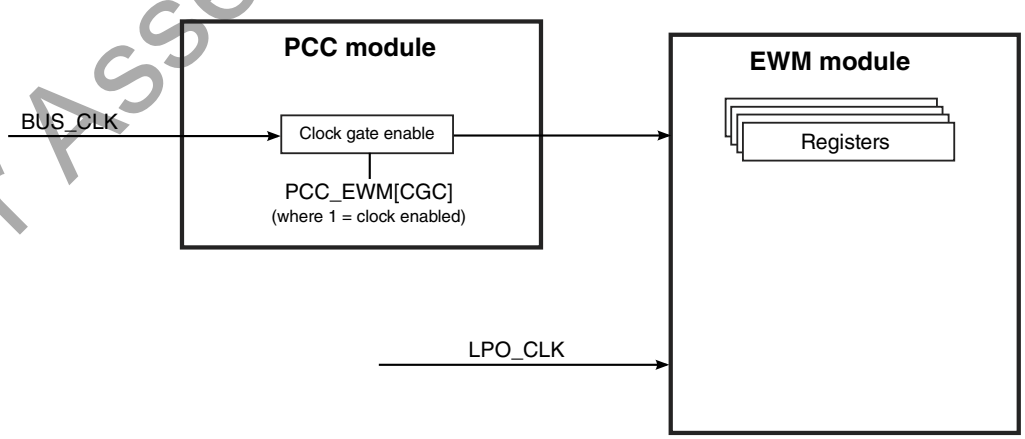


Figure 6-6. EWM clocking

For Assessment Purposes

6.6.3 WDOG clocks

The WDOG module uses the clocks as shown in the figure below.

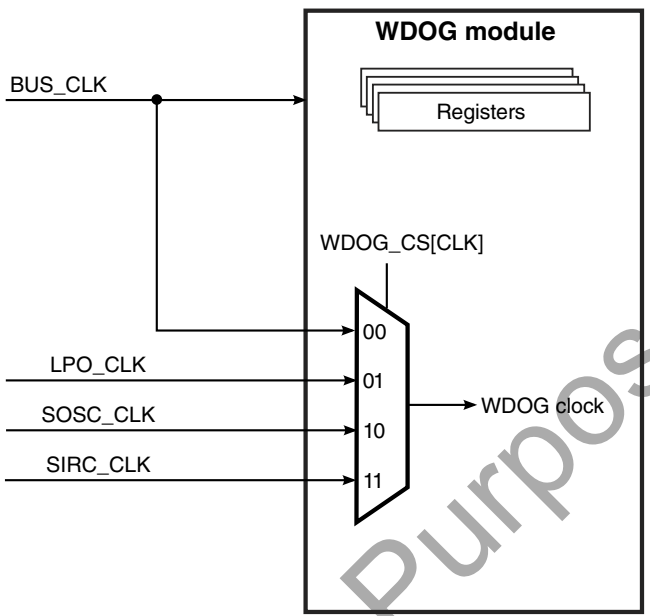


Figure 6-7. WDOG Clocking

6.6.4 FTFC clocks

The figure below the FTFC clocking configuration.

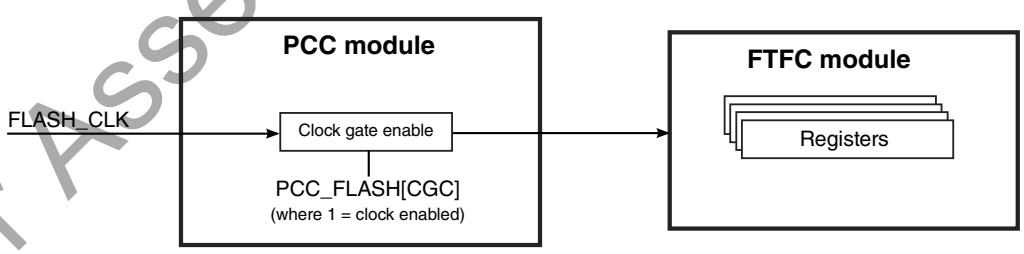


Figure 6-8. FTFC clocking

For Assessment Purposes Only

6.6.5 ADC conversion clock options

The ADC has multiple clock sources. Selection is determined by the configuration of PCC_ADCx[PCS]. The following table shows the available clock sources from the PCC module. The dividers should be configured such that the ADC conversion clock frequency lies within the valid range as per the ADC requirement (see the Data Sheet).

The following figure shows the clock sources available.

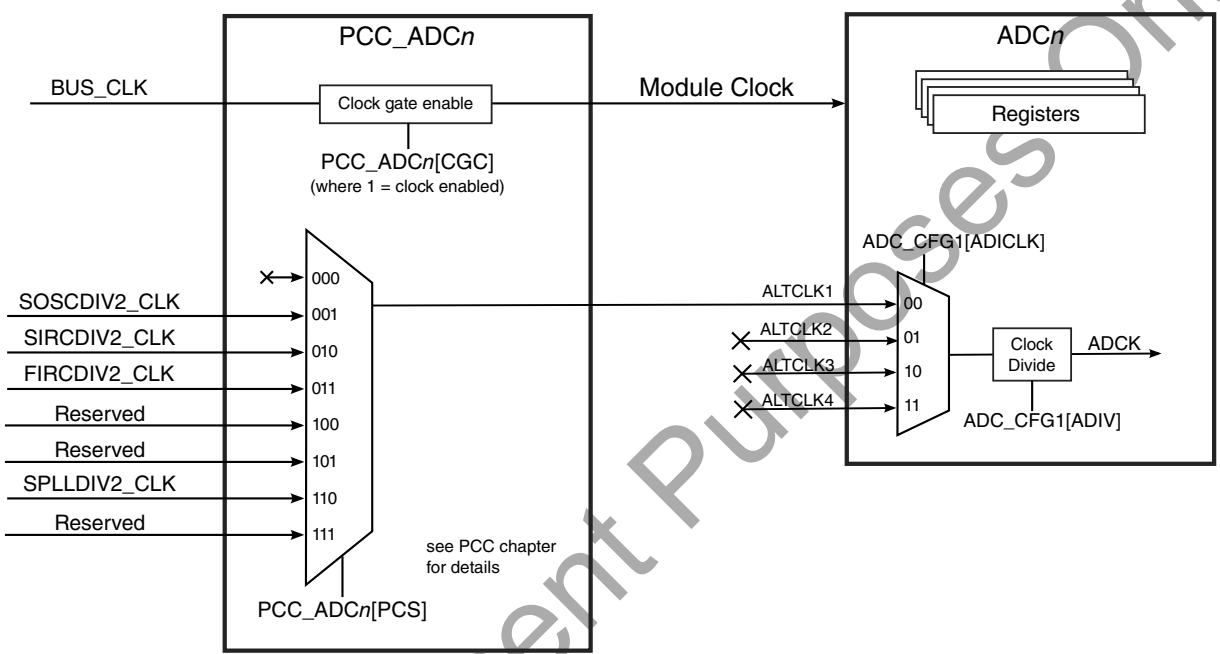


Figure 6-9. ADC Conversion Clock Options

6.6.6 PDB Clock Options

The PDB and FTM modules are clocked by the system clock (SYS_CLK). SYS_CLK provides higher timing resolution and more precise delay control in the PDB counter.

6.6.7 FTM Counter Clock Options

The FTM module is clocked by the internal SYS_CLK (the FTM module refers to it as system clock) which could be up to CPU frequency, but the FTM counter allows to be clocked by three clock sources:

- System clock (SYS_CLK) from PCC
- Internal fixed frequency clock
- External clocks from pins (TCLK[0:3])

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

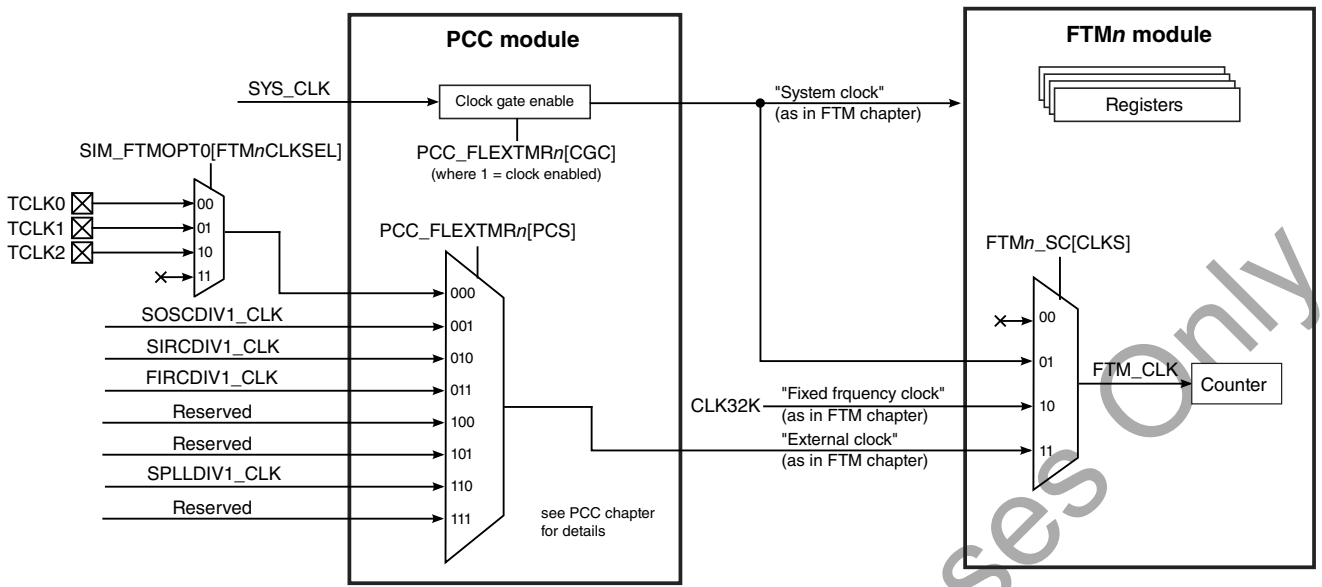


Figure 6-10. FTM Clocking

The counter clock source selection is controlled by CLKS bits in FTM*n*_SC register. The FTM_CLK is controlled by PCC module and could run up to CPU frequency and provide higher resolution for the FTM timer.

The fixed frequency clock is a fixed clock driven by CLK32K (RTC_CLKIN, LPO32K_CLK, SOSCDIV2_CLK, or FIRCDIV2_CLK). SIM_LPOCLKS[CLK32KSEL] is used to select the source of the CLK32K source.

There are three clocks that can be accessed externally to the chip (TCLK0, TCLK1, TCLK2). One of these clocks are selected to drive the 000b input of the multiplexer in the PCC (PCC_FLEXTMR*n*[PCS] = 000b) by configuring SIM_FTMOPT0[FTM*n*CLKSEL].

The fixed frequency clock and external clocks provide the user more clock options for FTM counter.

NOTE

Due to FTM hardware implementation limitations, the frequency of the fixed frequency clock must not exceed 1/2 of the FTM System clock frequency. Also, the frequency of the external clock source must not exceed 1/4 of FTM System clock frequency.

For Assessment Purposes Only

6.6.8 LPTMR prescaler/glitch filter clocking options

The prescaler and glitch filter of the LPTMR module can be clocked from misc sources determined by the PCC_LPTMR0[PCS] control bitfield and LPTMR0_PSR[PCS] bitfield. The LPTMR0_PSR[PCS] bitfield is used for internal LPTMR to select the clock source. The supported clock sources on this device are shown in Figure 6-11.

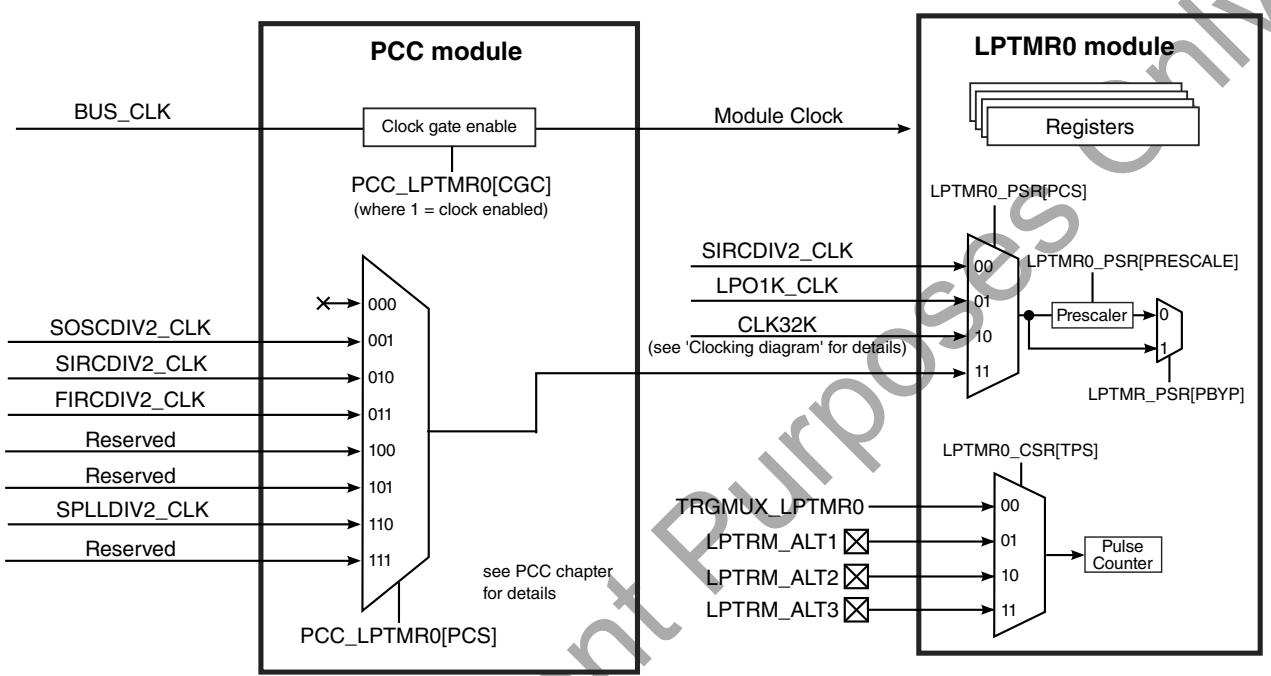


Figure 6-11. LPTMR clocks

NOTE

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

6.6.9 RTC_CLKOUT signal

RTC_CLKOUT is driven by the RTC, and is always enabled. There can be one of four source clocks that drive RTC_CLK as shown in Figure 6-1. The default clock source is SOSCDIV2_CLK.

6.6.10 FlexCAN Clocking

The figure below shows the FlexCAN module clocking diagram.

For Assessment Purposes Only

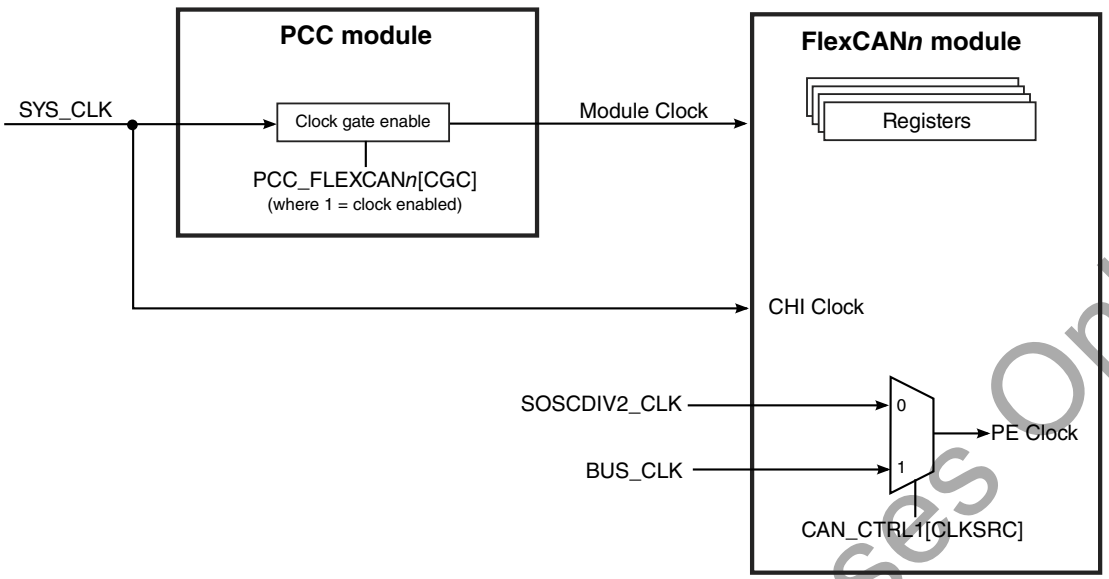


Figure 6-12. FlexCAN clocking

6.6.10.1 Clocking Options

CAN_CTRL1[CLKSRC] register bit selects between clocking the FlexCAN from the internal bus clock or the input clock (SOSCDIV2_CLK).

6.6.10.2 Clock Gating

The clock to each CAN module can be gated on and off using the PCC_FLEXCAN*n*[CGC] bits. These bits are cleared after any reset, which disables the clock to the corresponding module. The appropriate clock enable bit should be set by software at the beginning of the FlexCAN initialization routine to enable the module clock before attempting to initialize any of the FlexCAN registers.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 7
Reset and Boot

7.1 Introduction

The following reset sources are supported in this MCU:

Table 7-1. Reset sources

Reset sources	Description
POR reset	• Power-on reset (POR)
System resets	• External pin reset (PIN) • Low voltage detect (LVD) • Watchdog reset • Loss-of-clock (LOC) reset • Loss-of-lock (LOL) reset • Stop mode acknowledge error (SACKERR) • Software reset (SW) • Lockup reset (LOCKUP) • MDM DAP system reset
Debug reset	• JTAG reset

Each of the reset sources has an associated bit in the system reset status (RCM_SRS) register in Reset Control Module (RCM). Besides immediate reset, the RCM module also supports optional delays of the system resets for a period of time with an interrupt generated. This provides software an option to perform a graceful shutdown. See the Reset Control Module (RCM) chapter for register details. See the Reset Control Module for register details.

The MCU exits reset in functional mode where the CPU is executing code. See Boot options for more details.

For Assessment Purposes Only

7.2 Reset

This section 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 MCU or when the supply voltage drops below the power-on reset re-arm voltage level (V_{POR}), the 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_{LVD}). The POR and LVD bits in RCM_SRS register 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 regulator 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 are assigned by default to their analog functions after reset.

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

- TDI in pull-up (PU)
- TCK in pull-down (PD)
- TMS in PU

and associated output pin configured as:

- TDO with no pull-down or pull-up

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

7.2.2.1 External pin reset (PIN)

On this Device, RESET is a shared pin (Refer to the S32K144_IO_Signal_Description_Input_Multiplexing.xlsx attached to Reference Manual). The user needs to configure the corresponding PORT_PCR register prior to using this pin as RESET. On any reset event, the corresponding pad gets configured as RESET pin. Asserting RESET wakes the device from any mode. During a pin reset, the RCM_SRS[PIN] bit is set.

7.2.2.1.1 RESET pin filter

The RESET pin filter supports filtering from both the 128 kHz LPO clock and the bus clock. RCM_RPC[RSTFLTSS], RCM_RPC[RSTFLTSRW], and RCM_RPC[RSTFLTSEL] control this functionality; see the RCM chapter. The filters are asynchronously reset by Chip POR. The reset value for each filter assumes the RESET pin is negated.

For all stop modes where LPO clock is still active, 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.

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. The LVD system is always enabled in hsrun, normal run, or wait mode. The LVD system is disabled when entering VLPx modes or stop mode.

The LVD can be configured to generate a reset upon detection of a low voltage condition by setting the PMC_LVDSC1[LVDRE] bit to 1. 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. The RCM_SRS[LVD] bit is set following either an LVD reset or POR.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

7.2.2.3 Watchdog Reset

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 watchdog. If this periodic refreshing does not occur, the watchdog issues a system reset. The watchdog reset causes the RCM_SRS[WDOG] bit to set.

7.2.2.4 Loss-of-clock (LOC) reset

LOC reset is enabled when SOSC clock monitor is enabled (by setting SCG_SOSCCSR[SOSCCM]) and configured to generate a reset on error (by setting SCG_SOSCCSR[SOSCCMRE]). If the error bit (SCG_SOSCCSR[SOSCERR]) is set and LOC reset is enabled (as mentioned above), the MCU resets. The RCM_SRS[LOC] bit is set to indicate this reset source.

NOTE

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

7.2.2.5 Loss-of-lock (LOL) reset

The LOL reset is enabled when PLL clock monitor is enabled (by setting SCG_SPLLCSR[SPLLCM]) and configured to generate a reset on error (by setting SCG_SPLLCSR[SPLLCMRE]). If the error bit (SCG_SPLLCSR[SPLLERR]) is set and LOL reset is enabled (as mentioned above), the MCU resets. The RCM_SRS[LOL] bit is set to indicate this reset source.

NOTE

This reset source does not cause a reset if the chip is in any stop mode.

7.2.2.6 Stop mode acknowledge error (SACKERR)

This reset is generated if the core attempts to enter stop mode, but not all modules acknowledge stop mode within 1025 cycles of the 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.

The RCM_SRS[SACKERR] bit is set to indicate this reset source.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

7.2.2.7 Software reset (SW)

The SYSRESETREQ bit 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 the RCM_SRS[SW] bit 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 the RCM_SRS[LOCKUP] bit to set.

7.2.2.9 MDM-AP system reset request

Set the system reset request bit in the MDM-AP control register to initiate a system reset. This is the primary method for resets via the JTAG/SWD interface. The system reset is held until this bit is cleared.

Set the core hold reset bit 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 registers.

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

For Assessment Purposes Only

7.2.3.2 Chip POR

The Chip POR asserts on POR or LVD. It resets the Reset Pin Filter registers and parts of the SIM and SCG .

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

7.2.3.3 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).

7.2.3.4 Chip Reset

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

7.2.4 Reset Pin

For all reset sources, 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.

7.2.5 Debug resets

The following sections detail the debug resets available on the device.

7.2.5.1 JTAG reset

The JTAG module generate a system reset when certain IR codes are selected. This functional reset is asserted when EXTEST, HIGHZ and CLAMP instructions are active. The reset source from the JTAG module is released when any other IR code is selected. A JTAG reset causes the RCM_SRS[JTAG] bit to set.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

7.2.5.2 Resetting the Debug subsystem

Use the CDBGRSTREQ bit within the SWJ-DP CTRL/STAT register to reset the debug modules. However, as explained below, using the CDBGRSTREQ bit does not reset all debug-related registers.

CDBGRSTREQ resets the debug-related registers within the following modules:

- SWJ-DP
- AHB-AP
- TPIU
- MDM-AP (MDM control and status registers)

CDBGRSTREQ does not reset the debug-related registers within the following modules:

- CM4 core (core debug registers: DHCSR, DCRSR, DCRDR, DEMCR)
- FPB
- DWT
- ITM
- NVIC
- Crossbar bus switch
- Private peripheral bus

7.3 Boot

This section describes the boot sequence, including sources and options.

7.3.1 Boot sources

This device supports cold booting from internal flash.

When the device boots from internal flash, the reset vectors are located at address 0x0 (initial SP_main) and 0x4 (initial PC).

The device also supports boot from RAM by relocating the exception vector table to RAM. This is implemented through a programmable Vector Table Offset Register (VTOR) in NVIC module.

The device supports serial code download via CAN /LIN /SPI etc. The application developer can write their own flash based boot loaders or adapt the ones available at www.nxp.com. These boot loaders are then used to configure and use the communication

For Assessment Purposes Only

protocol to perform mass serial download. For instance, the CAN module is used to transfer data into its own message buffers, then the DMA module can be used to move the data into the main system RAM. Once the download is completed, the core can be used to transfer the new downloaded code into the flash area.

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 and any system reset. For more details on programming the option byte, see the flash memory chapter.

The MCU uses FTFE_FOPT to configure the device at reset as shown in the following table.

Purposes Only

Table 7-2. Flash Option Register (FTFE_FOPT) definition

Bit Num	Field	Value	Definition
7-6	Reserved	Reserved for future expansion.	
5	Reserved	Reserved	
3	RESET_PIN_CFG	Enables/disables control for the RESET pin.	
		0	RESET_b 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 pin low prior to establishing the setting of this option and releasing the reset function on the pin. This bit is preserved through system resets and low-power modes. When RESET_b 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	The port is configured with pullup enabled, 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 NMI_b pin controls with internal pullup enabled. When NMI_b pin function is disabled, it cannot be used as a source for low-power mode wake-up. If the NMI function is not required, either for an interrupt or wake up source, it is recommended that the NMI function be disabled by clearing NMI_DIS.
		1	NMI_b pin/interrupts reset default to enabled.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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

Bit Num	Field	Value	Definition
1	Reserved	Reserved for future expansion.	
0	LPBOOT	Controls the reset value of clock divider of IRC 48Mhz to feed the Core and platform clocks. Larger divide value selections produce lower average power consumption during POR and reset sequencing and after reset exit.	
		0	Core and system clock divider (OUTDIV1) is 0x1 (divide by 2).
		1	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 is above 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 Mode Controller reset logic then controls a sequence to exit reset.

1. A system reset is held on internal logic, the RESET_b pin is driven out low, and the SCG is enabled in its default clocking mode.
2. Required clocks are enabled (Core Clock, 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 Controller is released from reset and begins initialization operation while the Reset Control logic continues to drive the RESET_b pin out low.
4. Early in reset sequencing the NVM option byte is read and stored to the Flash Memory module's FOPT register. If the LPBOOT is 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_b pin is released. If RESET_b continues to be asserted (an indication of a slow rise time on the RESET_b pin or external drive in low), the system continues to be held in reset. Once the RESET_b 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.
7. If FlexNVM is enabled, the flash controller continues to restore the FlexNVM data. This data is not available immediately out of reset and the system should not access this data until the flash controller completes this initialization step as indicated by the EEERDY flag.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Subsequent system resets follow this same reset flow.

For Assessment Purposes Only

Chapter 8
Power Management

8.1 Introduction

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

8.2 Power Modes Description

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 Run and VLPR 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.

Stop mode entry is not supported directly from HSRUN and requires transition to Run prior to an attempt to enter a stop mode.

For Assessment Purposes Only

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

Table 8-1. Chip power modes

Chip mode	Description	Core mode	Normal recovery method
Normal run	Default mode out of reset; on-chip voltage regulator is on.	Run	-
High Speed run	Allows maximum performance of chip. In this state, the MCU is able to operate at a faster frequency compared to normal run mode.	Run	-
Normal Wait - via WFI	Allows peripherals to function while the core is in sleep mode, reducing power. NVIC remains sensitive to interrupts; peripherals continue to be clocked.	Sleep	Interrupt
Normal Stop - via WFI	Places chip in static state. Lowest power mode that retains all registers while maintaining LVD protection. NVIC is disabled; AWIC is used to wake up from interrupt; some peripheral clocks are stopped.	Sleep Deep	Interrupt
VLPR (Very Low Power Run)	On-chip voltage regulator is in a low power mode that supplies only enough power to run the chip at a reduced frequency. Reduced frequency Flash access mode (1 MHz); LVD off; SIRC or SOSC provides a low power 4 MHz source for the core, the bus and the peripheral clocks.	Run	-
VLPW (Very Low Power Wait) -via WFI	Same as VLPR but with the core in sleep mode to further reduce power; NVIC remains sensitive to interrupts (FLASH_CLK = ON). On-chip voltage regulator is in a low power mode that supplies only enough power to run the chip at a reduced frequency.	Sleep	Interrupt
VLPS (Very Low Power Stop)-via WFI	Places chip in static state with LVD operation off. Lowest power mode with ADC and pin interrupts functional. Some Peripheral clocks are stopped, but LPTMR, RTC, CMP can be used. NVIC is disabled (FLASH_CLK = OFF); AWIC is used to wake up from interrupt. On-chip voltage regulator is in a low power mode that supplies only enough power to run the chip at a reduced frequency. All SRAM is operating (content retained and I/O states held).	Sleep Deep	Interrupt

For Assess

8.3 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. The Nested Vectored Interrupt Controller (NVIC) describes interrupt operation and what peripherals can cause interrupts.

NOTE

The WFE (Wait for Event) 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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

8.4 Clocking modes

Following information describes the various clocking modes supported on this device.

8.4.1 Clock Gating

To conserve power, the clocks to most modules can be turned off using CGC bit of the peripheral control registers in the PCC module. These bits are cleared after any reset, which disables the Protocol clock to the corresponding module. Prior to initializing a module, set the corresponding bit in the PCC peripheral control register to enable the clock. Before turning off the clock, make sure to disable the module. For more details, refer to the chapters: Clock Distribution and PCC for details.

8.4.2 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, allowing the device to wakeup almost instantaneously; at the expense of higher power consumption. Partial Stop can be entered from either Run mode or VLP Run mode. There are two partial stop modes; PSTOP1 and PSTOP2.

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 SCG 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, the core clocks, system clocks and bus clocks are all gated.

8.4.3 DMA Wakeup

For Assessment Purposes Only

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

When a DMA wake-up 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 SCG, 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 wake-up 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 wake-up 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 wake-up negates and the DMA completes the current transfer, the device will transition back to 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, SCG 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 wake-up 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 wake-up completes, entry into the low power mode will restart.

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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

A DMA wake-up can be generated by either a synchronous DMA request or an asynchronous DMA request. 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.

NOTE

DMA Enable Asynchronous Request in Stop Register (DMA_EARS) is used to configure asynchronous DMA request for each channel.

8.4.4 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 Run mode, HSRUN 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. SCG, 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 bus (PPB) remains accessible during Compute Operation, including the MCM, System Control Space (SCS) (for NVIC), and SysTick. Although access to the GPIO registers 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

8.4.5 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 memory 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

If the flash memory 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 wake-up 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.

8.5 Power mode transitions

The following figure shows the power mode transitions. Any reset always brings the chip back to the normal run state. In run, wait, and stop modes active power regulation is enabled. The VLPx modes offer a lower power operating mode than normal modes. VLPR and VLPW are limited in frequency.

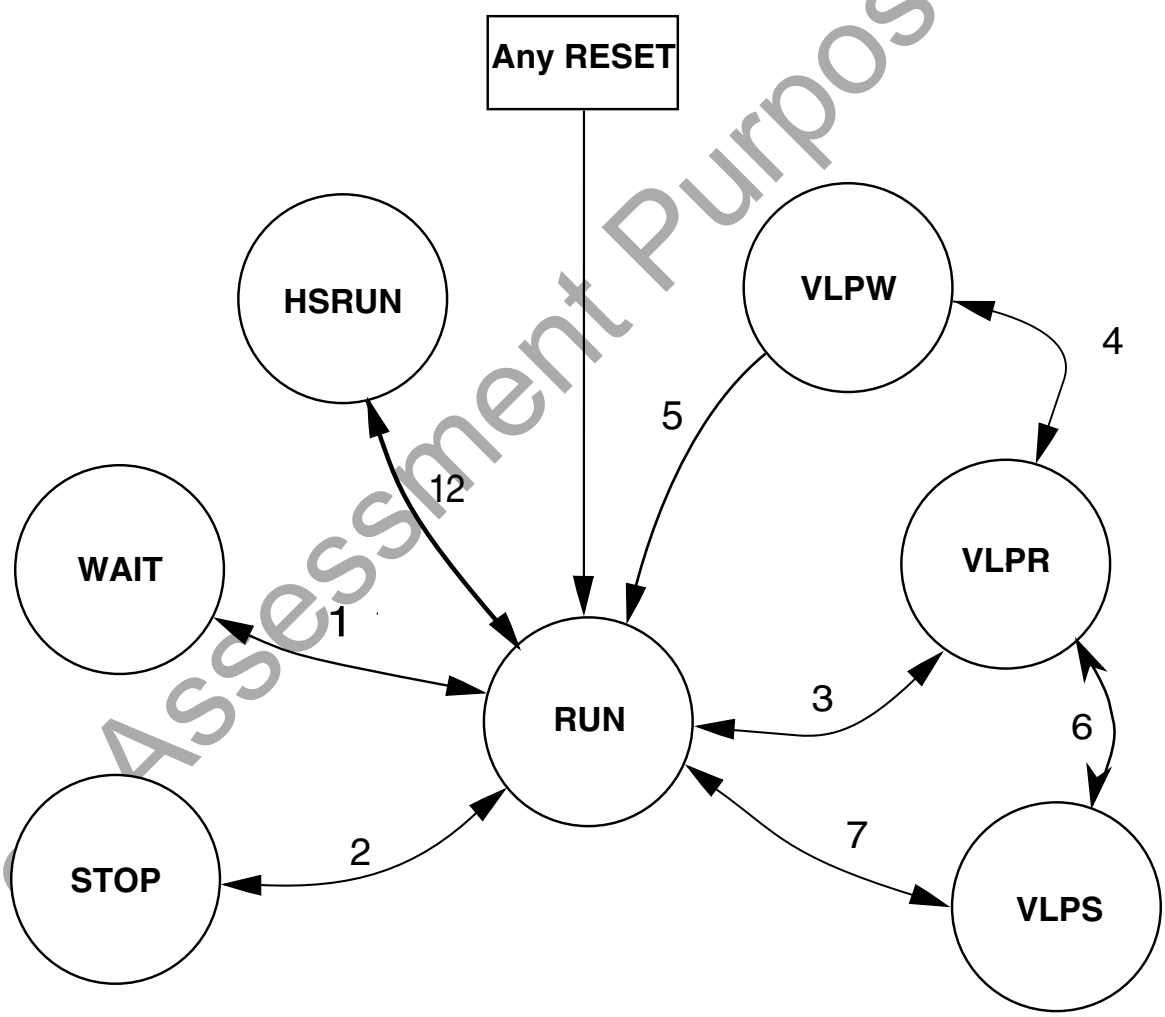


Figure 8-1. Power mode state transition diagram

For Assessment Purposes Only

NOTE

See Table 15-2 in the SMC chapter for more detailed mode transition conditions.

8.6 Power modes shutdown sequencing

When entering stop or other low-power modes, the clocks are shut off in an orderly sequence to safely place the chip in the targeted low-power state. All low-power entry sequences are initiated by the core executing an WFI instruction. The ARM core's outputs, SLEEPDEEP and SLEEPING, trigger entry to the various low-power modes:

- System level wait and VLPW modes equate to: SLEEPING & $\overline{\text{SLEEPDEEP}}$
- All other low power modes equate to: SLEEPING & SLEEPDEEP

When entering the non-wait modes, the chip performs the following sequence:

- Shuts off Core Clock and System Clock to the ARM Cortex-M4 core immediately.
- Polls stop acknowledge indications from the non-core crossbar masters (DMA), supporting peripherals (SPI, PIT) and the Flash Controller for indications that System Clocks, Bus Clock and/or Flash Clock need to be left enabled to complete a previously initiated operation, effectively stalling entry to the targeted low power mode. When all acknowledges are detected, System Clock, Bus Clock and Flash Clock are turned off at the same time.
- SCG and Mode Controller shut off clock sources and/or the internal supplies driven from the on-chip regulator as defined for the targeted low power mode.

In wait modes, most of the system clocks are not affected by the low power mode entry. The Core Clock to the ARM Cortex-M4 core is shut off. Some modules support stop-in-wait functionality and have their clocks disabled under these configurations.

The debugger modules support a transition from stop, wait, VLPS, and VLPW back to a halted state when the debugger is enabled. This transition is initiated by setting the Debug Request bit in MDM-AP control register. As part of this transition, system clocking is re-established and is equivalent to normal run/VLPR mode clocking configuration.

8.7 Flash Program Restrictions

The flash memory on this device should not be programmed or erased while operating in VLPR power mode.

For Assessment Purposes Only

8.8 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 2 MHz and 1 Mbit/s) 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.

Preliminary Purposes Only

Table 8-2. Module operation in low power modes

Modules	VLPR	VLPW	Stop	VLPS
Core modules				
NVIC	FF	FF	static	static
System modules				
Mode Controller	FF	FF	FF	FF
Regulator	low power	low power	low power	low power
LVD/LVR	disabled (LVR active only)	disabled (LVR active only)	disabled (LVR active only)	disabled (LVR active only)
Brown-out Detection	FF	FF	FF	FF
DMA	FF Async operation in CPO	FF	Async operation	Async operation
Watchdog	FF	FF	FF	FF
EWM	FF static in CPO	static	static FF in PSTOP2	static
Clocks				
128 kHz LPO	FF	FF	FF	FF
System oscillator (OSC)	OSC_CLK max of 40 MHz crystal	OSC_CLK max of 40 MHz crystal	OSC_CLK optional	OSC_CLK max of 40 MHz crystal

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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

Modules	VLPR	VLPW	Stop	VLPS
SCG	static - FIRC and SIRC optional; PLL optionally on but gated	SIRC	SIRC	static - no clock output
Core clock	4 MHz max	OFF	OFF	OFF
Platform clock	4 MHz max	4 MHz max	OFF	OFF
System clock	4 MHz max OFF in CPO	4 MHz max	OFF	OFF
Bus clock	4 MHz max OFF in CPO	4 MHz max	OFF 50 MHz max in PSTOP2 from RUN 4 MHz max in PSTOP2 from VLPR	OFF
Memory and memory interfaces				
Flash	1 MHz max access - no program/erase No register access in CPO	low power	low power	low power
System RAM (SRAM_U and SRAM_L)	low power[1]	low power	low power	low power
Cache	low power	low power	low power	low power
FlexMemory	low power [2]	low power	low power	low power
Communication interfaces				
LPUART	4 Mbps Async operation in CPO	4 Mbps	Async operation FF in PSTOP2	Async operation
LPSPI	1 Mbit/s (slave) 2 Mbit/s (master) static in CPO	1 Mbit/s (slave) 2 Mbit/s (master)	Async operation FF in PSTOP2	Async operation
LPI^{2}C	200 kbit/s static, address match wakeup in CPO	200 kbit/s	Async operation FF in PSTOP2	Async operation
FlexIO	FF Async operation in CPO	FF	Async operation FF in PSTOP2	Async opetaion
CAN	500 kbit/s wakeup in CPO	500 kbit/s	wakeup FF in PSTOP2	wakeup
Security				
CRC	FF static in CPO	FF	static	static
Timers				
FTM	FF static in CPO	FF	static	static
LPIT	FF	FF	Async operation	Async operation

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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

Modules	VLPR	VLPW	Stop	VLPS
	static in CPO		FF in PSTOP2	
PDB	FF static in CPO	FF	static	static
LPTMR	FF	FF	Async operation FF in PSTOP2	Async operation
Analog				
12-bit ADC	FF SIRC and OSC clocks only	FF SIRC and OSC clocks only	FF	FF SIRC and OSC clocks only
CMP [3]	LS compare only	LS compare only	HS or LS compare FF in PSTOP2	LS compare only
Human-machine interfaces				
GPIO	FF GPIO write only in CPO	FF	static output, wakeup input FF in PSTOP2	static output, wakeup input

1. SRAM is writable and readable in VLPR mode.
2. FlexRAM enabled as EEPROM is not writable in VLPR and writes are ignored. Read accesses to FlexRAM as EEPROM while in VLPR are allowed. There are no access restrictions for FlexRAM configured as traditional RAM.
3. CMP in stop or VLPS supports high speed or low speed external pin to pin or external pin to DAC compares. Windowed, sampled & filtered modes of operation are not available while in stop or VLPS modes.

For Assessment Purp

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 9
Security

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 the FSEC[SEC] bits. The MCU, in turn, confirms the security request and limits access to flash resources. During reset, the flash module initializes the FSEC register using data read from the security byte of the flash configuration field.

NOTE

The security features apply only to external accesses via debug. CPU accesses to the flash are not affected by the status of FSEC.

In the unsecured state all flash commands are available to the programming interfaces (SWD, JTAG), as well as user code execution of Flash Controller commands. When the flash is secured (FSEC[SEC] = 00, 01, or 11), programmer interfaces are only allowed to launch mass erase operations and have no access to memory locations.

Further information regarding the flash security options and enabling/disabling flash security is available in the Flash Memory Module.

For Assessment Purposes Only

9.3 Security Interactions with other Modules

The flash security settings are used by the SoC 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 JTAG port cannot access the memory resources of the MCU. Boundary scan chain operations work, but debugging capabilities are disabled so that the debug port cannot read flash contents.

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 bit in 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 10
Debug

10.1 Introduction

This device's debug is based on the ARM coresight architecture and is configured in each device to provide the maximum flexibility as allowed by the restrictions of the pinout and other available resources.

Several debug interfaces are supported:

- IEEE 1149.1 JTAG
- Serial Wire Debug (SWD)
- ARM Real-Time Trace Interface(1-pin asynchronous mode only)

The basic Cortex-M4 debug architecture is very flexible. The following diagram shows the topology of the core debug architecture and its components.

For Assessment Purposes Only

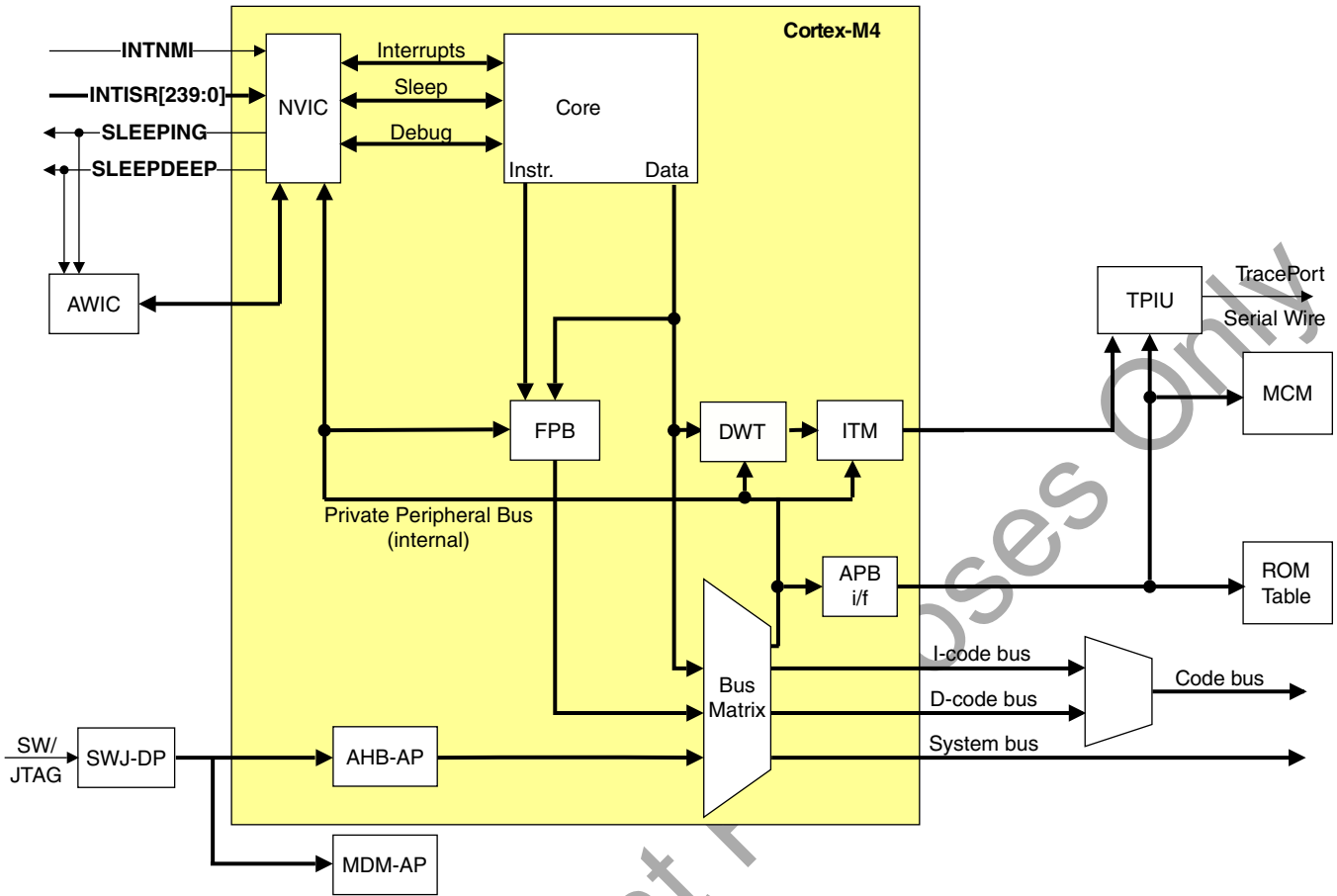


Figure 10-1. Cortex-M4 Debug Topology

The following table presents a brief description of each one of the debug components.

Table 10-1. Debug Components Description

Module	Description
SWJ-DP	Modified Debug Port with support for SWD, JTAG
AHB-AP	AHB Master Interface from JTAG to debug module and SOC system memory maps
MDM-AP	Provides centralized control and status registers for an external debugger to control the device.
ROM Table	Identifies which debug IP is available.
Core Debug	Singlestep, Register Access, Run, Core Status
ITM	S/W Instrumentation Messaging + Simple Data Trace Messaging + Watchpoint Messaging
DWT (Data and Address Watchpoints)	4 data and address watchpoints
FPB (Flash Patch and Breakpoints)	The FPB implements hardware breakpoints and patches code and data from code space to system space. The FPB unit contains two literal comparators for matching against literal loads from Code space, and remapping to a corresponding area in System space.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 10-1. Debug Components Description (continued)

Module	Description
	The FPB also contains six instruction comparators for matching against instruction fetches from Code space, and remapping to a corresponding area in System space. Alternatively, the six instruction comparators can individually configure the comparators to return a Breakpoint Instruction (BKPT) to the processor core on a match, so providing hardware breakpoint capability.
TPIU (Trace Port Inteface Unit)	Asynchronous Mode (1-pin) = TRACE_SWO (available on JTAG_TDO)

10.2 CM4 ROM Table

The ROM table is used to hold the information about the debug components.

1. The CM4 ROM table resides on the CM4 AHB AP and has entries for CM4 debug components.

Table 10-2. ROM Table Entries bit assignment

Bits	Name	Description
[31:12]	Address offset	Base address of the component, relative to the ROM address. Negative values are permitted using two's complement. ComponentAddress = ROMAddress + (AddressOffset SHL 12).
[11:2]	-	Reserved SBZ.
[1]	Format	1 = 32-bit format. In the DAP Debug ROM this is set to 1. 0 = 8-bit format.
[0]	Entry present	Set HIGH to indicate an entry is present.

Table 10-3. CM4 ROM Table

COMPONENT	ADDRESS	VALUE
NVIC	0xE00FF000	0xFFF0F003
DWT	0xE00FF004	0xFFF02003
FPB	0xE00FF008	0xFFF03003
ITM	0xE00FF00C	0xFFF01003
TPIU	0xE00FF010	0xFFF41003
ETM	0xE00FF014	0xFFF42002
ETB	0xE00FF018	0xFFF43002
FUNNEL	0xE00FF01C	0xFFF44002

Freescale Confidential Proprietary - Non-Disclosure Agreement required

10.3 The Debug Port

The configuration of the JTAG controller, and debug port is illustrated in the following figure:

NOTE

cJTAG module is not included on this device.

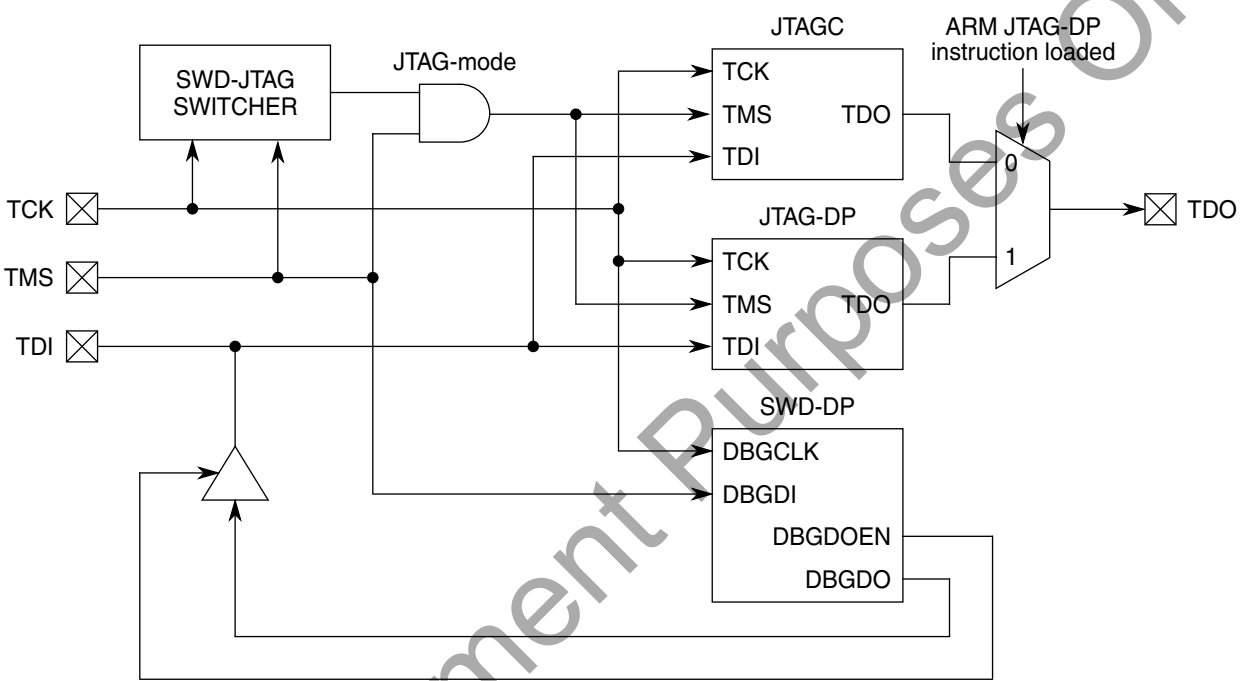


Figure 10-2. Modified Debug Port

The debug port comes out of reset in standard JTAG mode and is switched into SWD mode by the following sequences. Once the mode has been changed, unused debug pins can be reassigned to any of their alternative muxed functions.

10.3.1 JTAG-to-SWD change sequence

1. Send more than 50 TCK cycles with TMS (SWDIO) =1
2. Send the 16-bit sequence on TMS (SWDIO) = 0111_1001_1110_0111 (MSB transmitted first)
3. Send more than 50 TCK cycles with TMS (SWDIO) =1

NOTE

See the ARM documentation for the CoreSight DAP Lite for restrictions.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

10.4 Debug Port Pin Descriptions

The debug port pins default after POR to their JTAG functionality with the exception of JTAG_TRST_b and can be later reassigned to their alternate functionalities. In SWD mode, JTAG_TDI and JTAG_TRST_b can be configured to alternate GPIO functions.

Table 10-4. Debug port pins

Pin Name	JTAG Debug Port		SWD Debug Port		Internal Pull-up \Down
	Type	Description	Type	Description	
JTAG_TMS/ SWD_DIO	I/O	JTAG Test Mode Selection	I/O	Serial Wire Data	Pull-up
JTAG_TCLK/ SWD_CLK	I	JTAG Test Clock	I	Serial Wire Clock	Pull-down
JTAG_TDI	I	JTAG Test Data Input	-	-	Pull-up
JTAG_TDO/ TRACE_SWO	O	JTAG Test Data Output	O	Trace output over a single pin	N/C

10.5 System TAP connection

The system JTAG controller is connected in parallel to the ARM TAP controller. The system JTAG controller IR codes overlay the ARM JTAG controller IR codes without conflict. Refer to the IR codes table for a list of the available IR codes. The output of the TAPs (TDO) are muxed based on the IR code which is selected. This design is fully JTAG compliant and appears to the JTAG chain as a single TAP. At power on reset, ARM's IDCODE (IR=4'b1110) is selected.

10.5.1 IR Codes

Table 10-5. JTAG Instructions

Instruction	Code[3:0]	Instruction Summary
IDCODE	0000	Selects device identification register for shift
SAMPLE/PRELOAD	0010	Selects boundary scan register for shifting, sampling, and preloading without disturbing functional operation
SAMPLE	0011	Selects boundary scan register for shifting and sampling without disturbing functional operation

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 10-5. JTAG Instructions (continued)

Instruction	Code[3:0]	Instruction Summary
EXTEST	0100	Selects boundary scan register while applying preloaded values to output pins and asserting functional reset
HIGHZ	1001	Selects bypass register while three-stating all output pins and asserting functional reset
CLAMP	1100	Selects bypass register while applying preloaded values to output pins and asserting functional reset
ARM_IDCODE	1110	ARM JTAG-DP Instruction
BYPASS	1111	Selects bypass register for data operations
Factory debug reserved	0101, 0110, 0111, 1101	Intended for factory debug only
ARM JTAG-DP Reserved	1000, 1010, 1011, 1110	These instructions will go the ARM JTAG-DP controller. Please look at ARM JTAG-DP documentation for more information on these instructions.
Reserved [1]	All other opcodes	Decoded to select bypass register

1. The manufacturer reserves the right to change the decoding of reserved instruction codes in the future

10.6 JTAG 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 Figure 10-3. 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 (DAP) using JTAG or SWD. The MDM-AP is accessible as Debug Access Port 1 with the available registers shown in the table below.

ssessment Purpos

Table 10-6. 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	ID	Read-only identification register that always reads as 0x001C_0000

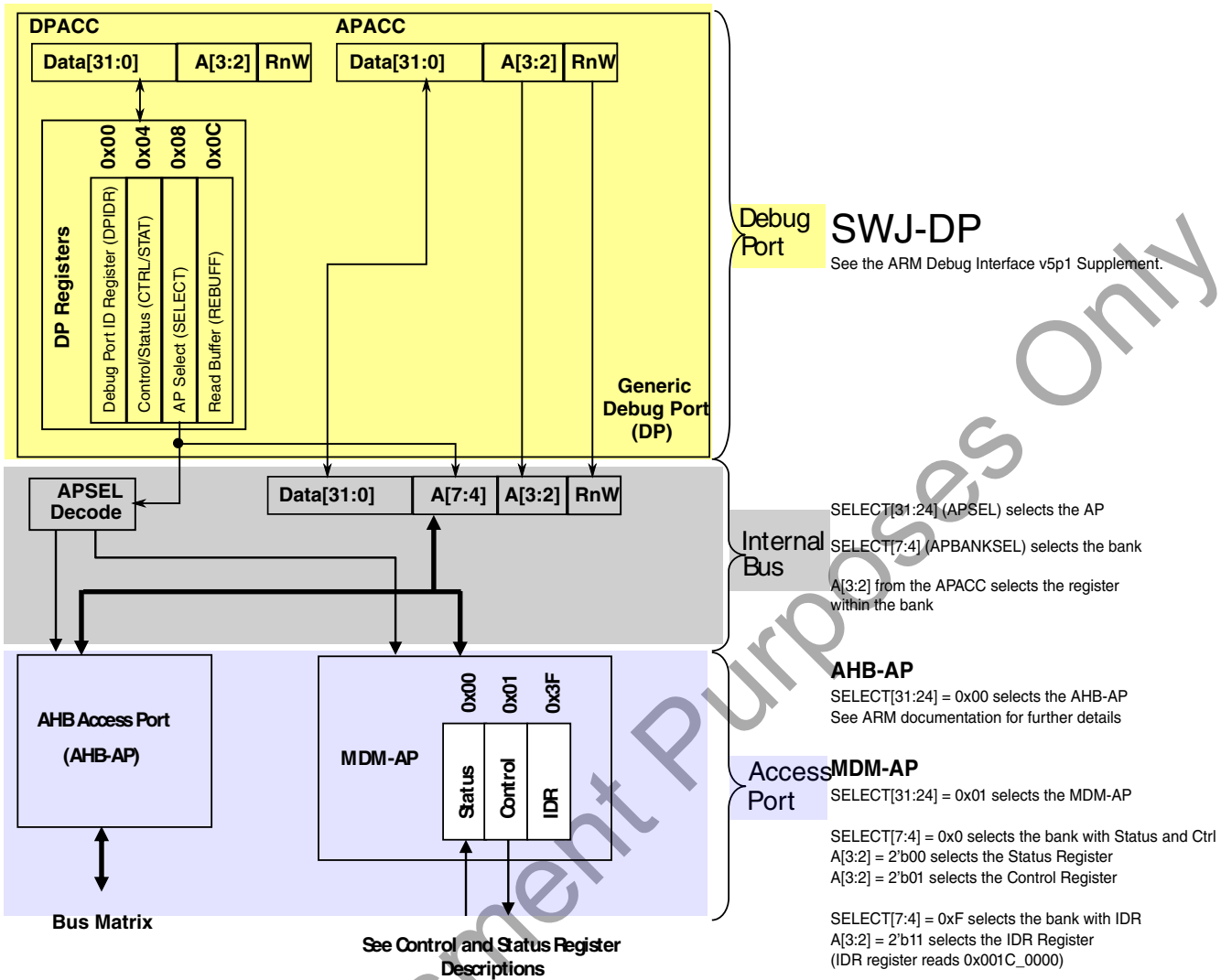


Figure 10-3. MDM AP Addressing

Assessment Purposes Only

10.6.1 MDM-AP Control Register

Table 10-7. MDM-AP Control register assignments

Bit	Name	Secure[1]	Description
0	Flash Mass Erase in Progress	Y	Set to cause mass erase. Cleared by hardware after mass erase operation completes. 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.
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.

Table continues on the next page...

Table 10-7. MDM-AP Control register assignments (continued)

Bit	Name	Secure[1]	Description
			If the Core is in a stop or wait mode, this bit can be used to wakeup the core and transition to a halted state.
3	System Reset Request	N	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 – 7	Reserved	N	
8	Timestamp Disable	N	Set this bit to disable the 48-bit global trace timestamp counter during debug halt mode when the core is halted. 0 The timestamp counter continues to count assuming trace is enabled. (default) 1 The timestamp counter freezes when the core has halted (debug halt mode).
9 – 31	Reserved for future use	N	

1. Command available in secure mode

10.6.2 MDM-AP Status Register

Table 10-8. MDM-AP Status register assignments

Bit	Name	Description
0	Flash Mass Erase Acknowledge	The Flash Mass Erase Acknowledge bit is cleared after any system 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	Indicate 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	

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 10-8. MDM-AP Status register assignments (continued)

Bit	Name	Description
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 [1] 0 Disabled 1 Enabled
7	LP Enabled	Decode of LPLLSM control bits to indicate that VLPS is the selected power mode the next time the ARM Core enters Deep Sleep. 0 Low Power Stop Mode is not enabled 1 Low Power Stop Mode is enabled 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 conjuntion 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 by debugger to throttle JTAG TCK frequency up/down.
9 – 10	Reserved	Always read 0.
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.

1. Refer to FTFE_FSEC[KEYEN] bit for more information.

For Assess

10.7 Debug Resets

The debug system receives the following sources of reset:

- Debug reset (CDBGRSTREQ bit within the SWJ-DP CTRL/STAT register) in the TCLK domain that allows the debugger to reset the debug logic.
- System POR reset

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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

10.8 AHB-AP

AHB-AP provides the debugger access to all memory and registers in the system, including processor registers through the NVIC. System access is independent of the processor status. AHB-AP does not do back-to-back transactions on the bus, so all transactions are non-sequential. AHB-AP can perform unaligned and bit-band transactions. AHB-AP transactions bypass the FPB, so the FPB cannot remap AHB-AP transactions. SWJ/SW-DP-initiated transaction aborts drive an AHB-AP-supported sideband signal called HABORT. This signal is driven into the Bus Matrix, which resets the Bus Matrix state, so that AHB-AP can access the Private Peripheral Bus for last ditch debugging such as read/stop/reset the core. AHB-AP transactions are little endian.

The MPU includes default settings and protections for the Region Descriptor 0 (RGD0) such that the Debugger always has access to the entire address space and those rights cannot be changed by the core or any other bus master.

For a short period at the start of a system reset event the system security status is being determined and debugger access to all AHB-AP transactions is blocked. The MDM-AP Status register is accessible and can be monitored to determine when this initial period is completed. After this initial period, if system reset is held via assertion of the RESET_b pin, the debugger has access via the bus matrix to the private peripheral bus to configure the debug IP even while system reset is asserted. While in system reset, access to other memory and register resources, accessed over the Crossbar Switch, is blocked.

10.9 ITM

The ITM is an application-driven trace source that supports printf style debugging to trace Operating System (OS) and application events, and emits diagnostic system information. The ITM emits trace information as packets. There are four sources that can generate packets. If multiple sources generate packets at the same time, the ITM arbitrates the order in which packets are output. The four sources in decreasing order of priority are:

1. Software trace -- Software can write directly to ITM stimulus registers. This emits packets.
2. Hardware trace -- The DWT generates these packets, and the ITM emits them.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

3. Time stamping -- Timestamps are emitted relative to packets. The ITM contains a 21-bit counter to generate the timestamp. The Cortex-M4 clock or the bitclock rate of the Serial Wire Viewer (SWV) output clocks the counter.
4. Global system timestamping. Timestamps can optionally be generated using a system-wide 48-bit count value.

10.10 Core Trace Connectivity

The ITM can route its data to the TPIU. (See the MCM (Miscellaneous Control Module) for controlling the routing to the TPIU.) This configuration enables the use of trace with low cost tools while maintaining the compatibility with trace probes.

10.11 TPIU

The TPIU acts as a bridge between the on-chip trace data from the Instrumentation Trace Macrocell (ITM) to a data stream, encapsulating IDs where required, that is then captured by a Trace Port Analyzer (TPA). The TPIU is specially designed for low-cost debug.

10.12 DWT

The DWT is a unit that performs the following debug functionality:

- It contains four comparators that you can configure as a hardware watchpoint, a PC sampler event trigger, or a data address sampler event trigger. The first comparator, DWT_COMP0, can also compare against the clock cycle counter, CYCCNT. The second comparator, DWT_COMP1, can also be used as a data comparator.
- The DWT contains counters for:
  - Clock cycles (CYCCNT)
  - Folded instructions
  - Load store unit (LSU) operations
  - Sleep cycles
  - CPI (all instruction cycles except for the first cycle)
  - Interrupt overhead

NOTE

An event is emitted each time a counter overflows.

- The DWT can be configured to emit PC samples at defined intervals, and to emit interrupt event information.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

10.13 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 are not allowed and 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 but also result in the system entering Stop mode instead.

10.13.1 Debug Module State in Low Power Modes

The following table shows the state of the debug modules in low power modes. 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.
- static = Module register states and associated memories are retained.
- OFF = Modules are powered off; module is in reset state upon wakeup.

Table 10-9. Debug Module State in Low Power Modes

Module	STOP	VLPR	VLPW	VLPS
Debug Port	FF	FF	FF	OFF
AHB-AP	FF	FF	FF	OFF
ITM	FF	FF	FF	OFF
TPIU	FF	FF	FF	OFF
DWT	FF	FF	FF	OFF

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

10.14 Debug & Security

When security is enabled (FSEC[SEC] != 10), 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 MDM-AP Status Register and can determine the current security state of the device. In the case of a secure device, the debugger also has the capability of performing a mass erase operation via writes to the MDM-AP Control Register. In the case of a secure device that has mass erase disabled (FSEC[MEEN] = 10), attempts to mass erase via the debug interface are blocked.

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 11
Signal Multiplexing and Pin Assignment

11.1 Introduction

The signal multiplexing module, IOMUX enables the sharing of single pad for multiple functions. The IOMUX consists only of combinatorial logic combined from several basic IOMUX cells. Each cell handles the signal multiplexing of only one pad.

The Port Control block controls the module specific pad settings (pull up etc) and the signal present on the external pin. Refer Port Control Register (PCR) for the description of control signals. For Reset values per port, see the S32K144_IO_Signal_Description_Input_Multiplexing.xlsx attached to the Reference Manual.

11.2 Functional Description

The IOMUX module handles the signal multiplexing of the device. The signal multiplexing architectural implementation is as shown in the following figure.

For Assessment Purposes Only

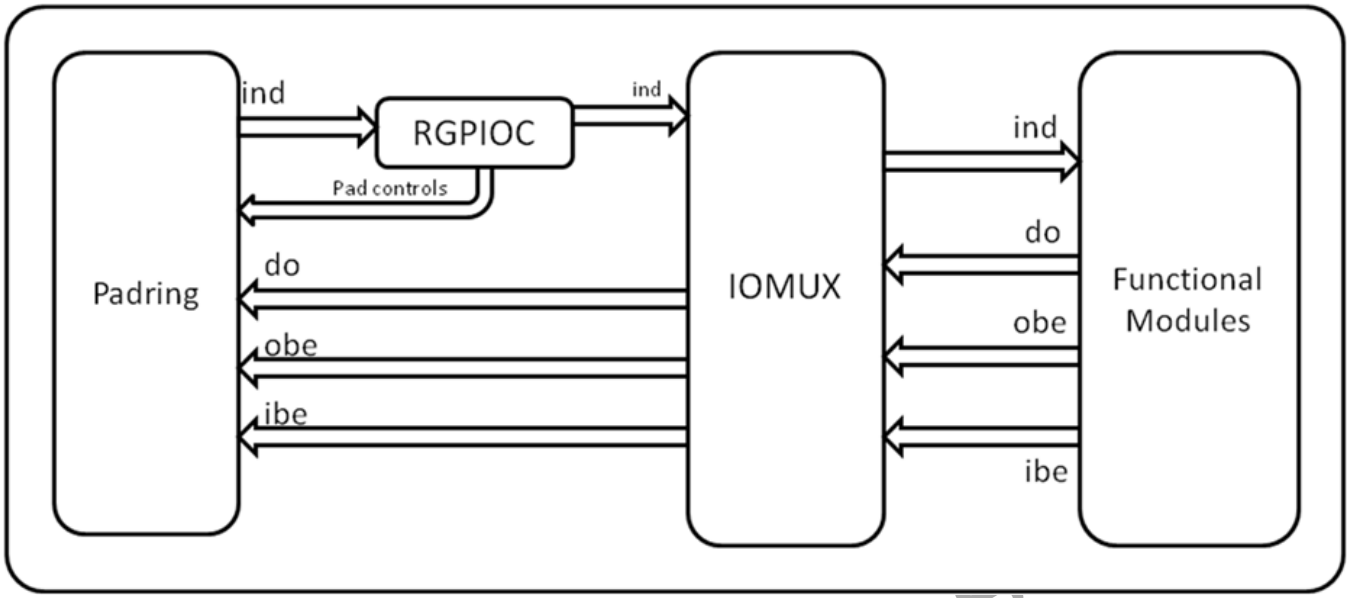


Figure 11-1. IOMUX architecture

11.3 Pad description

Following figure shows the basic representation of a GPIO Pad.

For Assessment Purpos

Freescale Confidential Proprietary - Non-Disclosure Agreement required

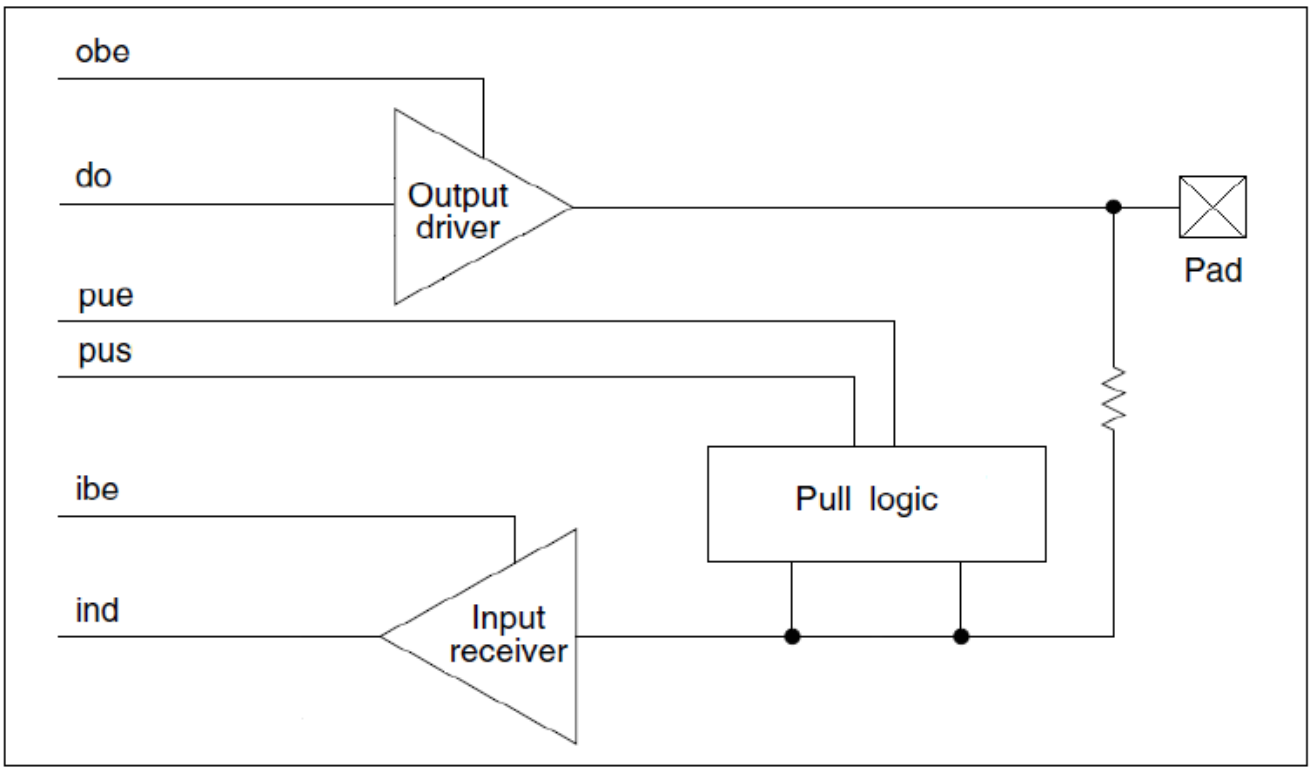


Figure 11-2. GPIO pad representation

Table 11-1. Pad Signal description

Signal name	Direction	Descriprtion
pad	I/O	I/O to external world
do	I	Data coming from the core into the pad
obe	I	Enable output driver
pue	I	0: Disable internal pullup or pulldown resistor 1: Enable internal pullup or pulldown resistor
pus	I	0: Enable internal pulldown resistor if pue is set 1: Enable internal pullup resistor if pue is set
ibe	I	Enable input receiver
ind	O	Data coming out of the pad into the core

Table 11-2. Truth table

obe	do	pad	Description
0	X	Z	Output buffer disabled, pad hi-Z (If not configured as input)
1	0/1	0/1	Output buffer enable, pad = do

pue	pus	pad	Description
0	X	-	Weak pull disabled. Pad retains previous state

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

pue	pus	pad	Description
1	0	0	Weak pull down enabled
1	1	1	Weak pull up enabled

ibe	pad	ind	Description
0	X	0	Input buffer disabled, ind gets low
1	0/1	0/1	Input buffer enabled, ind = pad

NOTE

The device does not support open drain on all the pins. Only pins that are configured for a protocol that requires open-drain (e.g;, IIC, LPUART single-wire) will work in open-drain mode.

11.4 IO Muxing sheet

The S32K144_IO_Signal_Description_Input_Multiplexing.xlsx sheet attached to the Reference Manual contains information on pins/balls of this device. Locate the paperclip symbol on the left side of the PDF window, and click it. Double-click on the Excel file to open it.

The excel file consists of several sheets. The 'IO Signal Table' and 'Input Muxing' tabs in the sheet correspond to the signal multiplexing information. The 'IO Signal Table' consists of all the pin muxing details and the 'Input Muxing' specifies the priority for the input muxing where an input path is driven by more than one pad.

11.4.1 IO Signal Table

Following is the snippet of IO Signal Table. For selecting any functionality, the pad PCR register needs to be configured accordingly.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Port	CR	SSS	Function	Module	Description	Direction	Pad Type	D1 KFA (100-qfp)	D1 KFA (64-qfp-10x10)	PCR										Reset	Bits Configurable
										ISF	IRQC[3:0]		LK	MUX[2:0]		DSE	ODE	PFE	SRE	PE	PS	
PTA0	PCR_PTA0	0000_0000	DISABLED		Signal Path Disabled	-	GPIO	79	50	0	0000		0	000		-	0	0	0	0	0	'00000000000000000000000000000000'	'11101100111000011111000010000000'
		0000_0001	PTA0	PTA		I/O																
		0000_0010	FTM_CH1	FTM2		I/O																
		0000_0011	LPI2C_SCLS	LPI2C0		I/O																
		0000_0100	FXIO_D2	FXIO		I/O																
		0000_0101	FTM_QD_PHA	FTM2		I																
		0000_0110	LPUART_CTS	LPUART0	Clear To Send (bar)	I																
		0000_0111	TRGMUX_OUT3	TRGMUX		O																
	-	-	ADC_SE0	ADC0		I																
	-	-	CMP_IN0	CMP0		I																

The columns of the above figure are described below:

- Port: This field in IO Signal Table specifies the PAD names of the device.
- CR(Control Register): This field specifies the name of PCR corresponding to the Port field. Refer PORT_PCR for description of PCR fields.
- SSS: This field specifies the ALT mode of operation as per PCR[Mux_mode]. Not all pins support all pin muxing slots. Unimplemented pin muxing slots are reserved. The corresponding pin is configured in the following pin muxing slot as follows:
  - 000: Alternative 0 (disabled/analog)
  - 001: Alternative 1 (GPIO)
  - 010: Alternative 2 (chip-specific)
  - 011: Alternative 3 (chip-specific)
  - 100: Alternative 4 (chip-specific)
  - 101: Alternative 5 (chip-specific)
  - 110: Alternative 6 (chip-specific)
  - 111: Alternative 7 (chip-specific)

  The analog functionalities are specified with '-' in this field. Here ADC_SE0 and CMP_IN0 represent analog functions.
- Function: This filed specifies the functionality of the pad as per the corresponding ALT mode specified by SSS field.
- Module: The Module field contains the module name of the VC which is governing the pad for the ALT mode.
- Description: This field mentions a short description of pad functionality.
- Direction: This filed specifies the direction (Input, Output or Inout) of the pad for the concerned functionality.
- The next two columns specify the pin number in 100 and 64 pin packages respectively.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- PCR: This field specifies the default PCR value for corresponding pad. Refer PORT_PCR for description of PCR fields.
- The next two columns specify the reset value and the configurable bit fields of PCR corresponding to pad.

11.4.2 Input muxing table

As the same function can be multiplexed to several pads, there can be a race condition if two pads have their PCR fields configured for same input path of a module. Here both the pads will start to drive the input path. To eliminate such condition, the input paths are prioritized. Following is a snippet of Input Muxing Table.

Destination Instance	Destination Function	Priority	SSS	Source Instance	Source Signal
CAN0	CAN_RX	1	0000_0101	IO_PAD	PTE4
		2	0000_0011	IO_PAD	PTC2
		3		-	disable low

The columns of the figure are briefly described below:

- Destination Instance: This field contains the instance name of the input path to where the signal will propagate from padring.
- Destination Function: This field mentions the function name of the input path.
- Priority: This field specifies the priority of the path. Priority level 1 is highest and it decreases onwards.
- SSS: This field specifies the PCR[Mux_mode] value corresponding to the pad specified in source signal column. A blank is mentioned for the default source when none pad is driving the input path.
- Source Instance: This field specifies the source pad type. A blank is mentioned for the default source when none pad is driving the input path.
- Source Signal: This field mentions the pad name. A ‘disable low’/’disable high’ specifies the connectivity when none of the pads are driving the input path.

11.5 Pinout diagrams

Refer I/O Signal Description and Input Multiplexing Tables Sheet for Pinout diagrams corresponding to 100 LQFP and 64 LQFP packages respectively.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 12
Port Control and Interrupts (PORT)

12.1 Chip specific PORT information

PCR bits corresponding to reset pad are non-sticky bits and on a functional reset, reset functionality on this pin will be resumed. Prior to entering any ALT functionality, PCR of corresponding pad should be properly configured.

12.2 Introduction

12.3 Overview

The Port Control and Interrupt (PORT) module provides support for port control, digital filtering, 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.

12.3.1 Features

The PORT module has the following features:

- Pin interrupt
  - 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

For Assessment Purposes Only

- Asynchronous wake-up in low-power modes
- Pin interrupt is functional in all digital pin muxing modes

- Digital input filter
  - Digital input filter for each pin, usable by any digital peripheral muxed onto the pin
  - Individual enable or bypass control field per pin
  - Selectable clock source for digital input filter with a five bit resolution on filter size
  - Functional in all digital pin multiplexing modes
- Port control
  - Individual pull control fields with pullup, pulldown, and pull-disable support
  - Individual drive strength field supporting high and low drive strength
  - Individual input passive filter field supporting enable and disable of the individual input passive filter
  - Individual mux control field supporting analog or pin disabled, GPIO, and up to six chip-specific digital functions
  - Pad configuration fields are functional in all digital pin muxing modes.

12.3.2 Modes of operation

12.3.2.1 Run mode

In Run mode, the PORT operates normally.

12.3.2.2 Wait mode

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

12.3.2.3 Stop mode

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

In Stop mode, the digital input filters are bypassed unless they are configured to run from the LPO clock source.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

12.3.2.4 Debug mode

In Debug mode, PORT operates normally.

12.4 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.

12.5 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.

12.6 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.

For Automotive Development Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PORT memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Pin Control Register n (PORT_PCR0)	32	R/W	See section	12.6.1/149
4	Pin Control Register n (PORT_PCR1)	32	R/W	See section	12.6.1/149
8	Pin Control Register n (PORT_PCR2)	32	R/W	See section	12.6.1/149
C	Pin Control Register n (PORT_PCR3)	32	R/W	See section	12.6.1/149
10	Pin Control Register n (PORT_PCR4)	32	R/W	See section	12.6.1/149
14	Pin Control Register n (PORT_PCR5)	32	R/W	See section	12.6.1/149
18	Pin Control Register n (PORT_PCR6)	32	R/W	See section	12.6.1/149
1C	Pin Control Register n (PORT_PCR7)	32	R/W	See section	12.6.1/149
20	Pin Control Register n (PORT_PCR8)	32	R/W	See section	12.6.1/149
24	Pin Control Register n (PORT_PCR9)	32	R/W	See section	12.6.1/149
28	Pin Control Register n (PORT_PCR10)	32	R/W	See section	12.6.1/149
2C	Pin Control Register n (PORT_PCR11)	32	R/W	See section	12.6.1/149
30	Pin Control Register n (PORT_PCR12)	32	R/W	See section	12.6.1/149
34	Pin Control Register n (PORT_PCR13)	32	R/W	See section	12.6.1/149
38	Pin Control Register n (PORT_PCR14)	32	R/W	See section	12.6.1/149
3C	Pin Control Register n (PORT_PCR15)	32	R/W	See section	12.6.1/149
40	Pin Control Register n (PORT_PCR16)	32	R/W	See section	12.6.1/149
44	Pin Control Register n (PORT_PCR17)	32	R/W	See section	12.6.1/149
48	Pin Control Register n (PORT_PCR18)	32	R/W	See section	12.6.1/149
4C	Pin Control Register n (PORT_PCR19)	32	R/W	See section	12.6.1/149
50	Pin Control Register n (PORT_PCR20)	32	R/W	See section	12.6.1/149
54	Pin Control Register n (PORT_PCR21)	32	R/W	See section	12.6.1/149
58	Pin Control Register n (PORT_PCR22)	32	R/W	See section	12.6.1/149
5C	Pin Control Register n (PORT_PCR23)	32	R/W	See section	12.6.1/149
60	Pin Control Register n (PORT_PCR24)	32	R/W	See section	12.6.1/149
64	Pin Control Register n (PORT_PCR25)	32	R/W	See section	12.6.1/149
68	Pin Control Register n (PORT_PCR26)	32	R/W	See section	12.6.1/149
6C	Pin Control Register n (PORT_PCR27)	32	R/W	See section	12.6.1/149
70	Pin Control Register n (PORT_PCR28)	32	R/W	See section	12.6.1/149
74	Pin Control Register n (PORT_PCR29)	32	R/W	See section	12.6.1/149
78	Pin Control Register n (PORT_PCR30)	32	R/W	See section	12.6.1/149
7C	Pin Control Register n (PORT_PCR31)	32	R/W	See section	12.6.1/149
80	Global Pin Control Low Register (PORT_GPCLR)	32	W (always reads 0)	0000_0000h	12.6.2/152
84	Global Pin Control High Register (PORT_GPCHR)	32	W (always reads 0)	0000_0000h	12.6.3/152
A0	Interrupt Status Flag Register (PORT_ISFR)	32	w1c	0000_0000h	12.6.4/153
C0	Digital Filter Enable Register (PORT_DFER)	32	R/W	0000_0000h	12.6.5/153

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PORT memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
C4	Digital Filter Clock Register (PORT_DFCR)	32	R/W	0000_0000h	12.6.6/154
C8	Digital Filter Width Register (PORT_DFWR)	32	R/W	0000_0000h	12.6.7/155

12.6.1 Pin Control Register n (PORT_PCR*n*)

NOTE

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

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

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

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

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

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

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

For Assessment Purposes Only

PORT_PCR*n* field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0 Configured interrupt is not detected. 1 Configured interrupt is detected. If the pin is configured to generate a DMA request, then the corresponding flag will be cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic 1 is written to the flag. If the pin is configured for a level sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt/DMA request as follows: 0000 Interrupt Status Flag (ISF) is disabled. 0001 ISF flag and DMA request on rising edge. 0010 ISF flag and DMA request on falling edge. 0011 ISF flag and DMA request on either edge. 0100 Reserved. 0101 Reserved. 0110 Reserved. 0111 Reserved. 1000 ISF flag and Interrupt when logic 0. 1001 ISF flag and Interrupt on rising-edge. 1010 ISF flag and Interrupt on falling-edge. 1011 ISF flag and Interrupt on either edge. 1100 ISF flag and Interrupt when logic 1. 1101 Reserved. 1110 Reserved. 1111 Reserved.
15 LK	Lock Register 0 Pin Control Register fields [15:0] are not locked. 1 Pin Control Register fields [15:0] are locked and cannot be updated until the next system reset.
14–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 MUX	Pin Mux Control Not all pins support all pin muxing slots. Unimplemented pin muxing slots are reserved and may result in configuring the pin for a different pin muxing slot. The corresponding pin is configured in the following pin muxing slot as follows: 000 Pin disabled (Alternative 0) (analog). 001 Alternative 1 (GPIO). 010 Alternative 2 (chip-specific).

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PORT_PCR*n* field descriptions (continued)

Field	Description
	011 Alternative 3 (chip-specific). 100 Alternative 4 (chip-specific). 101 Alternative 5 (chip-specific). 110 Alternative 6 (chip-specific). 111 Alternative 7 (chip-specific).
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 DSE	Drive Strength Enable Drive strength configuration is valid in all digital pin muxing modes. 0 Low drive strength is configured on the corresponding pin, if pin is configured as a digital output. 1 High drive strength is configured on the corresponding pin, if pin is configured as a digital output.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 PFE	Passive Filter Enable Passive filter configuration is valid in all digital pin muxing modes. 0 Passive input filter is disabled on the corresponding pin. 1 Passive input filter is enabled on the corresponding pin, if the pin is configured as a digital input. Refer to the device data sheet for filter characteristics.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0 Internal pullup or pulldown resistor is not enabled on the corresponding pin. 1 Internal pullup or pulldown resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0 Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1 Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

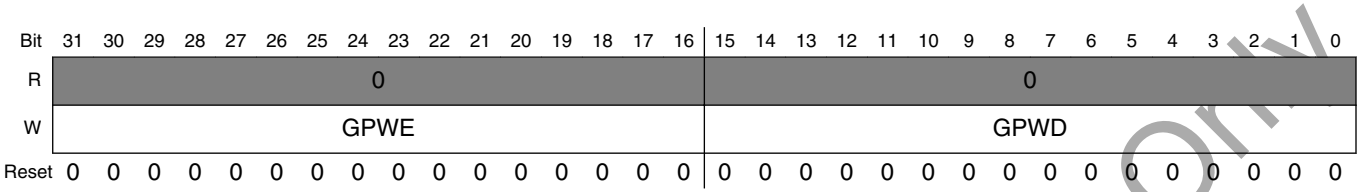
For

Freescale Confidential Proprietary - Non-Disclosure Agreement required

12.6.2 Global Pin Control Low Register (PORT_GPCLR)

Only 32-bit writes are supported to this register.

Address: 0h base + 80h offset = 80h



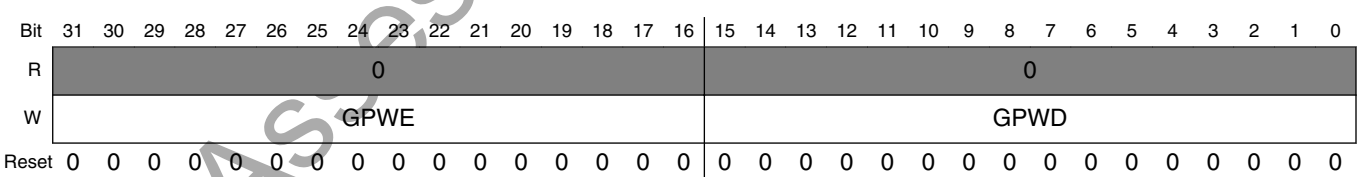
PORT_GPCLR field descriptions

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

12.6.3 Global Pin Control High Register (PORT_GPCHR)

Only 32-bit writes are supported to this register.

Address: 0h base + 84h offset = 84h



PORT_GPCHR field descriptions

Field	Description
31–16 GPWE	Global Pin Write Enable Selects which Pin Control Registers (31 through 16) bits [15:0] update with the value in GPWD. If a selected Pin Control Register is locked then the write to that register is ignored. 0 Corresponding Pin Control Register is not updated with the value in GPWD. 1 Corresponding Pin Control Register is updated with the value in GPWD.
GPWD	Global Pin Write Data

Table continues on the next page...

Assessment Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PORT_GPCHR field descriptions (continued)

Field	Description
	Write value that is written to all Pin Control Registers bits [15:0] that are selected by GPWE.

12.6.4 Interrupt Status Flag Register (PORT_ISFR)

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

Address: 0h base + A0h offset = A0h

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

PORT_ISFR field descriptions

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

12.6.5 Digital Filter Enable Register (PORT_DFER)

The digital filter configuration is valid in all digital pin muxing modes.

Address: 0h base + C0h offset = C0h

Bit	31	30	29	28	27	26	25	24	23	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	DFE																															
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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PORT_DFER field descriptions

Field	Description
DFE	Digital Filter Enable The digital filter configuration is valid in all digital pin muxing modes. The output of each digital filter is reset to zero at system reset and whenever the digital filter is disabled. Each bit in the field enables the digital filter of the same number as the field. 0 Digital filter is disabled on the corresponding pin and output of the digital filter is reset to zero. 1 Digital filter is enabled on the corresponding pin, if the pin is configured as a digital input.

12.6.6 Digital Filter Clock Register (PORT_DFCR)

The digital filter configuration is valid in all digital pin muxing modes.

Address: 0h base + C4h offset = C4h

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															CS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORT_DFCR field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 CS	Clock Source The digital filter configuration is valid in all digital pin muxing modes. Configures the clock source for the digital input filters. Changing the filter clock source must be done only when all digital filters are disabled. 0 Digital filters are clocked by the bus clock. 1 Digital filters are clocked by the LPO clock.

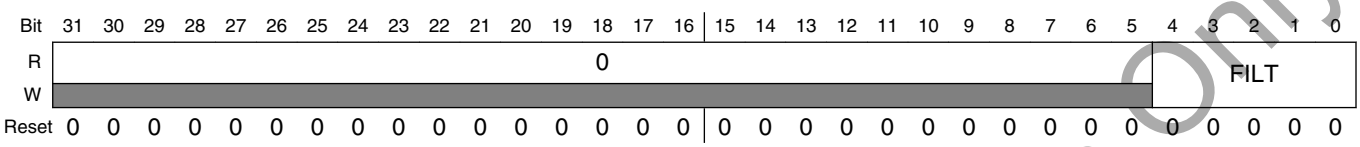
For Development Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

12.6.7 Digital Filter Width Register (PORT_DFWR)

The digital filter configuration is valid in all digital pin muxing modes.

Address: 0h base + C8h offset = C8h



PORT_DFWR field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FILT	Filter Length The digital filter configuration is valid in all digital pin muxing modes. Configures the maximum size of the glitches, in clock cycles, that the digital filter absorbs for the enabled digital filters. Glitches that are longer than this register setting will pass through the digital filter, and glitches that are equal to or less than this register setting are filtered. Changing the filter length must be done only after all filters are disabled.

12.7 Functional description

12.7.1 Pin control

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

The upper half of the Pin Control register configures the pin's capability to either interrupt the CPU or request a DMA transfer, on a rising/falling edge or both edges as well as a logic level occurring on the port pin. It also includes a flag to indicate that an interrupt has occurred.

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

- Pullup or pulldown enable
- Drive strength
- Passive input filter enable
- Pin Muxing mode

For Assessment Only

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 LK bit (bit 15 of Pin Control Register PCR*n*) allows the configuration for each pin to be locked until the next system reset. When locked, writes to the lower half of that pin control register are ignored, although a bus error is not generated on an attempted write to a locked register.

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.

12.7.2 Global pin control

The two global pin control registers allow a single register write to update the lower half of the pin control register on up to 16 pins, all with the same value. Registers that are locked cannot be written using the global pin control registers.

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

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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

12.7.3 External interrupts

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

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

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

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

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

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

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

For Assessment Purposes Only

12.7.4 Digital filter

The digital filter capabilities of the PORT module are available in all digital Pin Muxing modes if the PORT module is enabled.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The clock used for all digital filters within one port can be configured between the bus clock or the LPO clock. This selection must be changed only when all digital filters for that port are disabled. If the digital filters for a port are configured to use the bus clock, then the digital filters are bypassed for the duration of Stop mode. While the digital filters are bypassed, the output of each digital filter always equals the input pin, but the internal state of the digital filters remains static and does not update due to any change on the input pin.

The filter width in clock size is the same for all enabled digital filters within one port and must be changed only when all digital filters for that port are disabled.

The output of each digital filter is logic zero after system reset and whenever a digital filter is disabled. After a digital filter is enabled, the input is synchronized to the filter clock, either the bus clock or the LPO clock. If the synchronized input and the output of the digital filter remain different for a number of filter clock cycles equal to the filter width register configuration, then the output of the digital filter updates to equal the synchronized filter input.

The maximum latency through a digital filter equals three filter clock cycles plus the filter width configuration register.

For Assessment Purposes Only

Chapter 13
System Integration Module (SIM)

13.1 Introduction

The System Integration Module (SIM) provides system control and chip configuration registers.

13.1.1 Features

Features of the SIM include:

- System clocking configuration
- Flash and system RAM size configuration
- FlexTimer clock and channel selection and configuration
- ADC trigger selection
- LPO clock source selection
- Flash configuration
- System device identification (ID)

13.2 Memory map and register definition

The SIM module contains many fields for selecting the clock source and dividers for various module clocks. See the Clock Distribution chapter for more information, including block diagrams and clock definitions.

NOTE

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

For Assessment Purposes Only

SIM memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4	Chip Control register (SIM_CHIPCTL)	32	R/W	See section	13.2.1/161
C	FTM Option Register 0 (SIM_FTMOPT0)	32	R/W	0000_0000h	13.2.2/163
10	LPO Clock Select Register (SIM_LPOCLKS)	32	R/W	0000_0003h	13.2.3/165
18	ADC Options Register (SIM_ADCOPT)	32	R/W	0000_0000h	13.2.4/166
1C	FTM Option Register 1 (SIM_FTMOPT1)	32	R/W	0000_0000h	13.2.5/168
24	System Device Identification Register (SIM_SDID)	32	R	Undefined	13.2.6/170
40	Platform Clock Gating Control Register (SIM_PLATCGC)	32	R/W	0000_001Fh	13.2.7/171
4C	Flash Configuration Register 1 (SIM_FCFG1)	32	R	Undefined	13.2.8/172
50	Flash Configuration Register 2 (SIM_FCFG2)	32	R	See section	13.2.9/175
54	Unique Identification Register High (SIM_UIDH)	32	R	Undefined	13.2.10/176
58	Unique Identification Register Mid-High (SIM_UIDMH)	32	R	Undefined	13.2.11/176
5C	Unique Identification Register Mid Low (SIM_UIDML)	32	R	Undefined	13.2.12/177
60	Unique Identification Register Low (SIM_UIDL)	32	R	Undefined	13.2.13/177
68	System Clock Divider Register 4 (SIM_CLKDIV4)	32	R/W	1000_0000h	13.2.14/178
6C	Miscellaneous Control register (SIM_MISCTRL)	32	R/W	0000_0000h	13.2.15/179

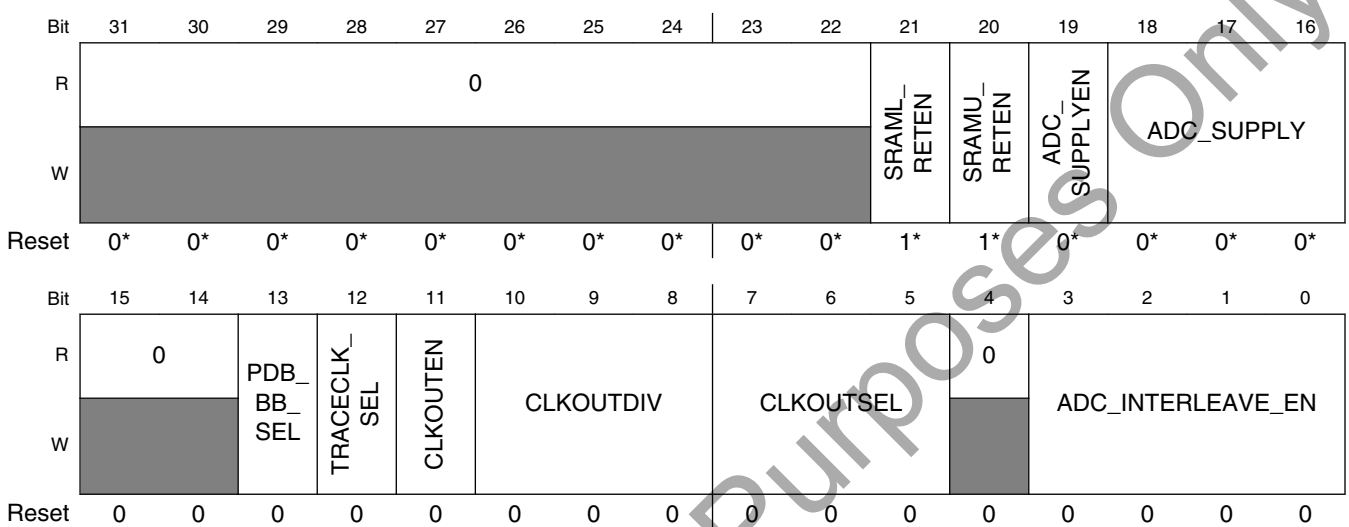
For Assessment Pur

Freescale Confidential Proprietary - Non-Disclosure Agreement required

13.2.1 Chip Control register (SIM_CHIPCTL)

SIM_CHIPCTL contains the controls for selecting ADC COCO trigger, trace clock, clock out source, PDB back-to-back mode and ADC interleave channel.

Address: 0h base + 4h offset = 4h



* Notes:

- Reserved field: This bit is reset on POR
- SRAML_RETEN field: This bit is reset on POR
- SRAMU_RETEN field: This bit is reset on POR
- ADC_SUPPLYEN field: This bit is reset on POR
- ADC_SUPPLY field: This bit is reset on POR

SIM_CHIPCTL field descriptions

Field	Description
31–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 SRAML_RETEN	SRAML retention 0 SRAM contents are retained across resets 1 No SRAM retention
20 SRAMU_RETEN	SRAMU retention 0 SRAM contents are retained across resets 1 No SRAM retention
19 ADC_ SUPPLYEN	Enable for internal supply monitoring on ADC channel AD21. 0 disable internal supply monitoring 1 enable internal supply monitoring
18–16 ADC_SUPPLY	Internal supplies monitored on ADC channel AD21.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ment Purposes Only

SIM_CHIPCTL field descriptions (continued)

Field	Description
	000 5V input VDD supply (VDD) 001 5V input analog supply (VDDA) 010 ADC Reference Supply (VREFH) 011 3.3V Oscillator Regulator Output (VDD_3V) 100 3.3V flash regulator output (VDD_flash_3V) 101 1.2V core regulator output (VDD_LV) 110 Reserved 111 Reserved
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 PDB_BB_SEL	PDB back-to-back select Selects ADC COCO source as pdb back-to-back mode, see Back-to-back acknowledgement connections for details. 0 PDB0 channel 0 back-to-back operation with ADC0 COCO[7:0]; PDB1 channel 0 back-to-back operation with ADC1 COCO[7:0] 1 Channel 0 of PDB0,PDB1 back-to-back operation with COCO[7:0] of ADC0, ADC1 .
12 TRACECLK_SEL	Debug trace clock select Selects core clock or platform clock as the trace clock source. 0 core clock 1 platform clock
11 CLKOUTEN	Enable for Clockout 0 Clockout disable 1 Clockout enable
10–8 CLKOUTDIV	Clockout Divide Ratio 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
7–5 CLKOUTSEL	CLKOUT Select Selects the clock to output on the CLKOUT pin. 000 SCG CLKOUT 001 SOSC DIV2 CLK 010 SIRC DIV2 CLK 011 FIRC DIV2 CLK 100 SPLL DIV2 CLK 101 LPO CLK 128 Khz

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_CHIPCTL field descriptions (continued)

Field	Description
	110 LPO CLK as selected by SIM_LPOCLKS[LPOCLKSEL] 111 RTC CLK as selected by SIM CLK 32 KHz Select
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ADC_ INTERLEAVE_ EN	ADC interleave channel enable Select ADC interleave pins. Bit 3 to 0 are for PTB14, PTB13, PTB1 and PTB0 respectively. 0000 No interleave channel Bit 3: PTB14 to ADC1_SE9 and ADC0_SE9 Bit 2: PTB13 to ADC1_SE8 and ADC0_SE8 Bit 1: PTB1 to ADC0_SE5 and ADC1_SE15 Bit 0: PTB0 to ADC0_SE4 and ADC1_SE14

13.2.2 FTM Option Register 0 (SIM_FTMOPT0)

Address: 0h base + Ch offset = Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FTM3CLKSEL		FTM2CLKSEL		FTM1CLKSEL		FTM0CLKSEL		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	FTM3FLTxSEL			0	FTM2FLTxSEL			0	FTM1FLTxSEL			0	FTM0FLTxSEL		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_FTMOPT0 field descriptions

Field	Description
31–30 FTM3CLKSEL	FTM3 External Clock Pin Select Selects the external pin used to drive the clock to the FTM3 module. **NOTE:** The selected pin must also be configured for the FTM external clock function through the appropriate Pin Control Register in the Port Control module. 00 FTM3 external clock driven by TCLK0 pin. 01 FTM3 external clock driven by TCLK1 pin. 10 FTM3 external clock driven by TCLK2 pin. 11 No clock input
29–28 FTM2CLKSEL	FTM2 External Clock Pin Select Selects the external pin used to drive the clock to the FTM2 module. **NOTE:** The selected pin must also be configured for the FTM external clock function through the appropriate Pin Control Register in the Port Control module.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_FTMOPT0 field descriptions (continued)

Field	Description
	00 FTM2 external clock driven by TCLK0 pin. 01 FTM2 external clock driven by TCLK1 pin. 10 FTM2 external clock driven by TCLK2 pin. 11 No clock input
27–26 FTM1CLKSEL	FTM1 External Clock Pin Select Selects the external pin used to drive the clock to the FTM1 module. **NOTE:** The selected pin must also be configured for the FTM external clock function through the appropriate Pin Control Register in the Port Control module. 00 FTM1 external clock driven by TCLK0 pin. 01 FTM1 external clock driven by TCLK1 pin. 10 FTM1 external clock driven by TCLK2 pin. 11 No clock input
25–24 FTM0CLKSEL	FTM0 External Clock Pin Select Selects the external pin used to drive the clock to the FTM0 module. **NOTE:** The selected pin must also be configured for the FTM external clock function through the appropriate Pin Control Register in the Port Control module. 00 FTM0 external clock driven by TCLK0 pin. 01 FTM0 external clock driven by TCLK1 pin. 10 FTM0 external clock driven by TCLK2 pin. 11 No clock input
23–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–12 FTM3FLTxSEL	FTM3 Fault X Select Selects the source of FTM3 fault. Every bit means one fault input respectively. **NOTE:** The pin source for fault must be configured for the FTM module fault function through the appropriate pin control register in the port control module when it comes from external fault pin. TRGMUX_FTM3 SELx is corresponding to FTM3 Fault x input . Bit value = 0: FTM3_FLTx pin Bit value = 1: TRGMUX_FTM3 out
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 FTM2FLTxSEL	FTM2 Fault X Select Selects the source of FTM2 fault. Every bit means one fault input respectively. **NOTE:** The pin source for fault must be configured for the FTM module fault function through the appropriate pin control register in the port control module when it comes from external fault pin. TRGMUX_FTM2 SELx is corresponding to FTM2 Fault x input . Bit value = 0: FTM2_FLTx pin Bit value = 1: TRGMUX_FTM2 out
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_FTMOPT0 field descriptions (continued)

Field	Description
6–4 FTM1FLTxSEL	FTM1 Fault X Select Selects the source of FTM1 fault. Every bit means one fault input respectively. **NOTE:** The pin source for fault must be configured for the FTM module fault function through the appropriate pin control register in the port control module when it comes from external fault pin. TRGMUX_FTM1 SELx is corresponding to FTM1 Fault x input. Bit value = 0: FTM1_FLTx pin Bit value = 1: TRGMUX_FTM1 out
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FTM0FLTxSEL	FTM0 Fault X Select Selects the source of FTM0 fault. Every bit means one fault input respectively. **NOTE:** The pin source for fault must be configured for the FTM module fault function through the appropriate pin control register in the port control module when it comes from external fault pin. TRGMUX_FTM0 SELx is corresponding to FTM0 Fault x input. Bit value = 0: FTM0_FLTx pin Bit value = 1: TRGMUX_FTM0 out

13.2.3 LPO Clock Select Register (SIM_LPOCLKS)

NOTE

The LPOCLKS register is a write-once register, and only reset on POR or LVD.

Address: 0h base + 10h offset = 10h

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										CLK32KSEL		LPOCLKSEL		LPO32KCLKEN	LPO1KCLKEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

For Assessment Pu

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_LPOCLKS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 CLK32KSEL	32 kHz clock source select Selects 32 kHz clock source for peripherals 00 SOSCDIV2_CLK 01 32 kHz LPO clock 10 RTC_CLKIN clock 11 FIRCDIV2_CLK
3–2 LPOCLKSEL	LPO clock source select Selects LPO clock source for peripherals 00 128 kHz LPO clock 01 no clock 10 32 kHz LPO clock which is derived from the 128 kHz LPO clock 11 1 kHz LPO clock which is derived from the 128 kHz LPO clock
1 LPO32KCLKEN	32 kHz LPO clock enable 0 Disable 32 kHz LPO clock output 1 Enable 32 kHz LPO clock output
0 LPO1KCLKEN	1 kHz LPO clock enable 0 Disable 1 kHz LPO clock output 1 Enable 1 kHz LPO clock output

13.2.4 ADC Options Register (SIM_ADCOPT)

Address: 0h base + 18h offset = 18h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		ADC1PRETRGSEL		ADC1SWPRETRG			ADC1TRGSEL	0		ADC0PRETRGSEL		ADC0SWPRETRG			ADC0TRGSEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_ADCOPT field descriptions

Field	Description
31–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–12 ADC1PRETRGSEL	ADC1 pre-trigger source select Selects pre-trigger source for ADC1. 00 PDB pre-trigger (default) 01 TRGMUX pre-trigger 10 Software pre-trigger 11 Reserved
11–9 ADC1SWPRETRG	ADC1 software pre-trigger sources 000 software pre-trigger disabled 001 - 011 Reserved (do not use) 100 software pre-trigger 0 101 software pre-trigger 1 110 software pre-trigger 2 111 software pre-trigger 3
8 ADC1TRGSEL	ADC1 trigger source select Selects trigger source for ADC1. **NOTE:** Each PDB supports two ADC channels, and each channel is with 8 pre-triggers. The trigger of two ADC channels are OR'ed together to support up to 16 pre-triggers if necessary. 0 PDB output 1 TRGMUX output
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 ADC0PRETRGSEL	ADC0 pre-trigger source select Selects pre-trigger source for ADC0. 00 PDB pre-trigger (default) 01 TRGMUX pre-trigger 10 Software pre-trigger 11 Reserved
3–1 ADC0SWPRETRG	ADC0 software pre-trigger sources 000 software pre-trigger disabled 001 - 011 Reserved (do not use) 100 software pre-trigger 0 101 software pre-trigger 1 110 software pre-trigger 2 111 software pre-trigger 3

Table continues on the next page...

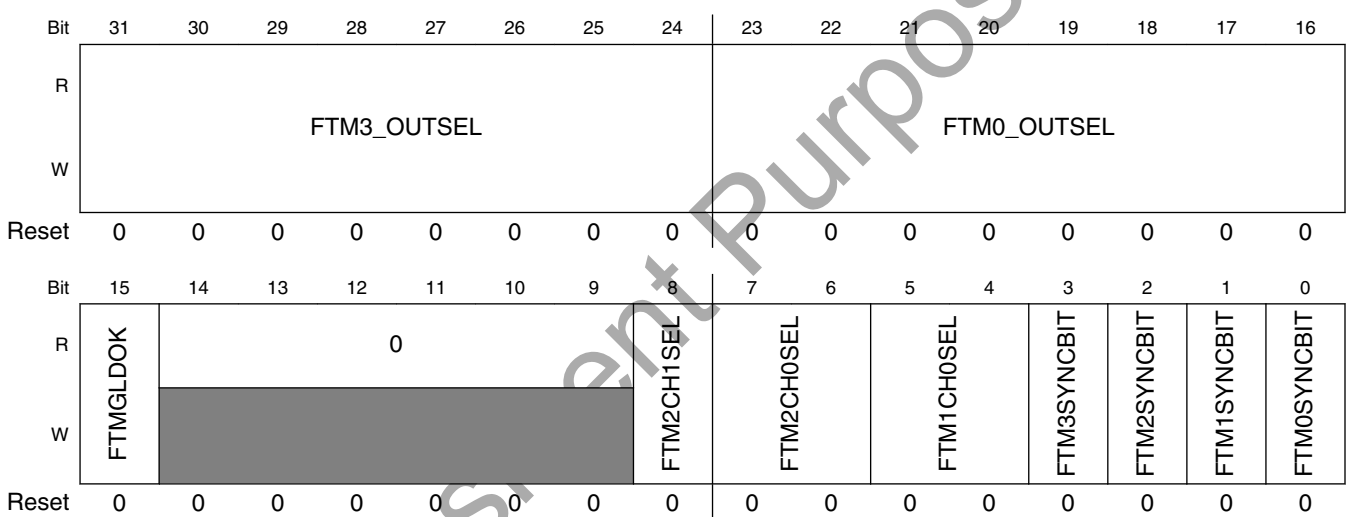
Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_ADCOPT field descriptions (continued)

Field	Description
0 ADC0TRGSEL	ADC0 trigger source select Selects trigger source for ADC0. **NOTE:** Each PDB supports two ADC channels, and each channel is with 8 pre-triggers. The trigger of two ADC channels are OR'ed together to support up to 16 pre-triggers if necessary. 0 PDB output 1 TRGMUX output

13.2.5 FTM Option Register 1 (SIM_FTMOPT1)

Address: 0h base + 1Ch offset = 1Ch



SIM_FTMOPT1 field descriptions

Field	Description
31–24 FTM3_OUTSEL	FTM3 channel modulation select with FTM2_CH1 Bit 7 to 0 are for channel 7 to 0 respectively. 0 No modulation with FTM2_CH1 1 Modulation with FTM2_CH1
23–16 FTM0_OUTSEL	FTM0 channel modulation select with FTM1_CH1 Bit 7 to 0 are for channel 7 to 0 respectively. 0 No modulation with FTM1_CH1 1 Modulation with FTM1_CH1
15 FTMGLDOK	FTM global load enable FTM global load enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_FTMOPT1 field descriptions (continued)

Field	Description
	0 FTM Global load mechanism disabled. 1 FTM Global load mechanism enabled
14–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 FTM2CH1SEL	FTM2 CH1 Select Selects FTM2 CH1 input 0 FTM2_CH1 input 1 exclusive OR of FTM2_CH0,FTM2_CH1, and FTM1_CH1
7–6 FTM2CH0SEL	FTM2 CH0 Select Selects FTM2 CH0 input 00 FTM2_CH0 input 01 CMP0 output 10 Reserved 11 Reserved
5–4 FTM1CH0SEL	FTM1 CH0 Select Selects FTM1 CH0 input 00 FTM1_CH0 input 01 CMP0 output 10 Reserved 11 Reserved
3 FTM3SYNCBIT	FTM3 Sync Bit
2 FTM2SYNCBIT	FTM2 Sync Bit
1 FTM1SYNCBIT	FTM1 Sync Bit
0 FTM0SYNCBIT	FTM0 Sync Bit

For

Freescale Confidential Proprietary - Non-Disclosure Agreement required

13.2.6 System Device Identification Register (SIM_SDID)

NOTE

Reset value loaded during System Reset from Flash IFR.

Address: 0h base + 24h offset = 24h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GENERATION				SUBSERIES				DERIVATE				RAMSIZE				REVID				PACKAGE				FEATURES							
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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_SDID field descriptions

Field	Description
31–28 GENERATION	S32K Family Generation Specifies the generation of the S32K family of the devices. Generation for this device is 1.
27–24 SUBSERIES	Subseries Specifies the S32K sub-series of the device. Sub-series for this device is 4.
23–20 DERIVATE	Derivate Specifies the derivate of the device. Derivate of this device is 4
19–16 RAMSIZE	RAM size This field specifies the amount of system RAM available on the device. 0xF: 64 KB 0xD: 48 KB
15–12 REVID	Device revision number Specifies the silicon implementation number for the device. REV ID for this device is 0.
11–8 PACKAGE	Package Specifies the available package options for the device. 0x3: 64 LQFP 0x4: 100 LQFP 0x8: 100 MAP BGA
FEATURES	Features Specifies the supported features of the device. 1: Feature is present

Table continues on the next page...

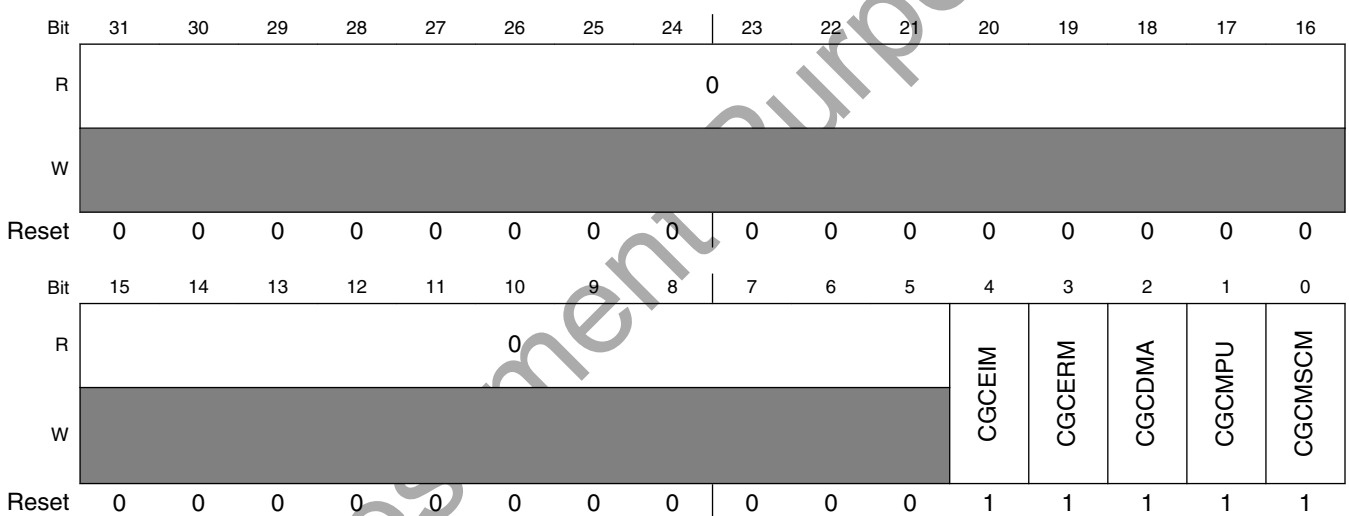
Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_SDID field descriptions (continued)

Field	Description
	0: Feature is not present Bit 7: Security Bit 6: ISO CAN-FD Bit 5: FlexIO Bit 4: Reserved Bit 3: Reserved Bit 2: Reserved Bit 1: Reserved Bit 0: Reserved

13.2.7 Platform Clock Gating Control Register (SIM_PLATCGC)

Address: 0h base + 40h offset = 40h



SIM_PLATCGC field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 CGCEIM	EIM Clock Gating Control Controls the clock gating to the EIM module. 0 Clock disabled 1 Clock enabled
3 CGCERM	ERM Clock Gating Control Controls the clock gating to the ERM module.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_PLATCGC field descriptions (continued)

Field	Description
	0 Clock disabled 1 Clock enabled
2 CGCDMA	DMA Clock Gating Control Controls the clock gating to the DMA module. 0 Clock disabled 1 Clock enabled
1 CGCMPU	MPU Clock Gating Control Controls the clock gating to the MPU module. 0 Clock disabled 1 Clock enabled
0 CGCMSCM	MSCM Clock Gating Control Controls the clock gating to the MSCM module. 0 Clock disabled 1 Clock enabled

13.2.8 Flash Configuration Register 1 (SIM_FCFG1)

NOTE

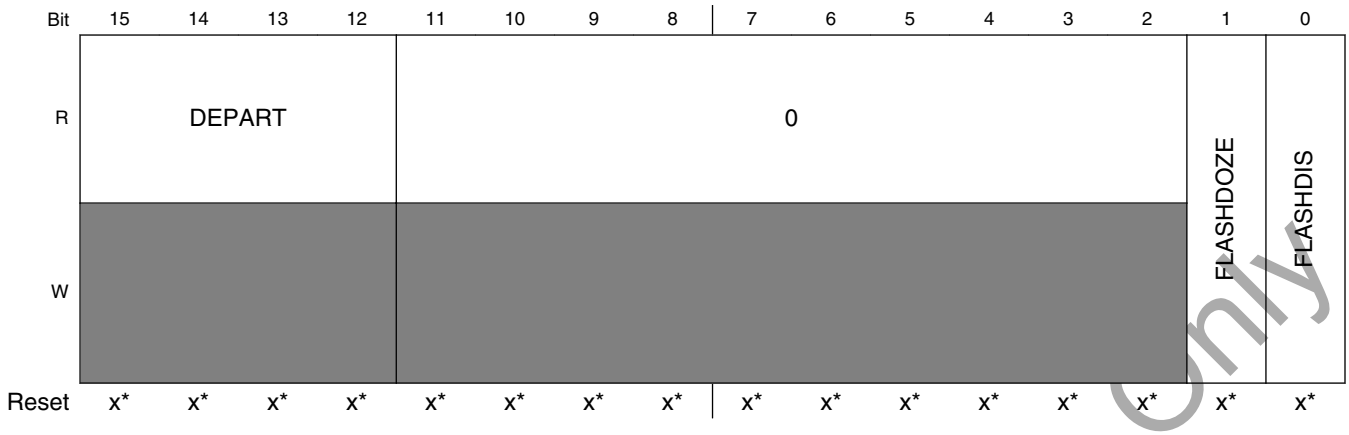
Reset value of NVMSIZE, PFSIZE, EEERAM_SIZE, DEPART loaded during System Reset from Flash IFR.

Address: 0h base + 4Ch offset = 4Ch

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

Freescale Assessment Purpose

Freescale Confidential Proprietary - Non-Disclosure Agreement required



* Notes:
- x = Undefined at reset.
- x = Undefined at reset.

ses Only

SIM_FCFG1 field descriptions

Field	Description
31–28 NVMSIZE	FlexNVM size This field specifies the amount of FlexNVM memory available on the device. Undefined values are reserved. 0000 0 KB of FlexNVM 0011 32 KB of FlexNVM 0101 64 KB of FlexNVM 0111 128 KB of FlexNVM 1001 256 KB of FlexNVM 1111 512 KB of FlexNVM
27–24 PFSIZE	Program flash size This field specifies the amount of program flash memory available on the device . Undefined values are reserved. 0000 8 KB of program flash memory, 0.25 KB protection region 0001 16 KB of program flash memory, 0.5 KB protection region 0011 32 KB of program flash memory, 1 KB protection region 0101 64 KB of program flash memory, 2 KB protection region 0111 128 KB of program flash memory, 4 KB protection region 1001 256 KB of program flash memory, 8 KB protection region 1011 512 KB of program flash memory, 16 KB protection region 1111 512 KB of program flash memory, 16 KB protection region
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 EEERAMSIZE	EEE SRAM SIZE EEE SRAM data size . 0000 Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SIM_FCFG1 field descriptions (continued)

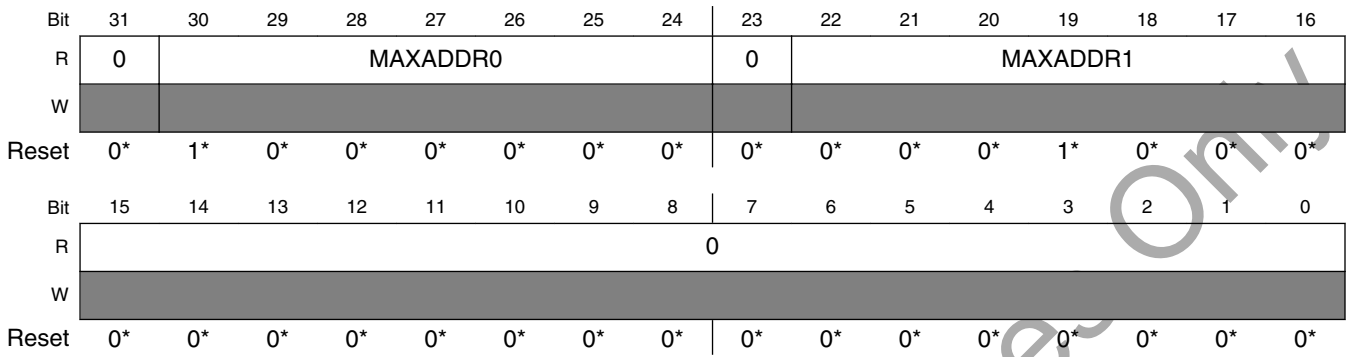
Field	Description
	0001 Reserved 0010 4 KB 0011 2 KB 0100 1 KB 0101 512 Bytes 0110 256 Bytes 0111 128 Bytes 1000 64 Bytes 1001 32 Bytes 1010-1110 Reserved 1111 0 Bytes
15–12 DEPART	FlexNVM partition Data flash / EEPROM backup split . See DEPART bit description in FTFE chapter.
11–2 Reserved	This field 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 Wait mode. An attempt by the DMA or other bus master to access the Flash when the Flash is disabled will result in a bus error. This bit should be clear during VLP modes. The Flash will be automatically enabled again at the end of Wait mode so interrupt vectors do not need to be relocated out of Flash memory. The wakeup time from Wait mode is extended when this bit is set. 0 Flash remains enabled during Wait mode 1 Flash is disabled for the duration of Wait 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 bit 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

For Ass

Freescale Confidential Proprietary - Non-Disclosure Agreement required

13.2.9 Flash Configuration Register 2 (SIM_FCFG2)

Address: 0h base + 50h offset = 50h



* Notes:
- Reset values are applicable for no EEE partition.

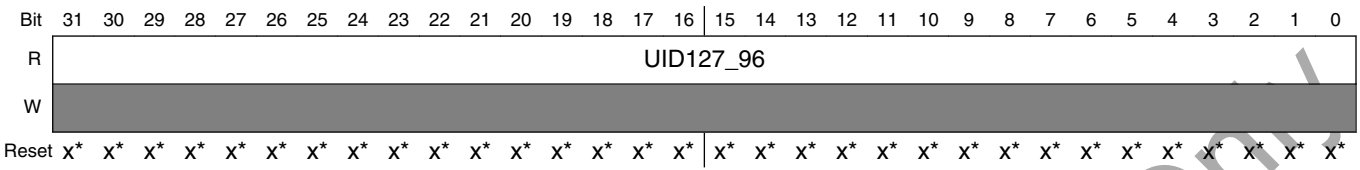
SIM_FCFG2 field descriptions

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

For Internal Use Only

13.2.10 Unique Identification Register High (SIM_UIDH)

Address: 0h base + 54h offset = 54h



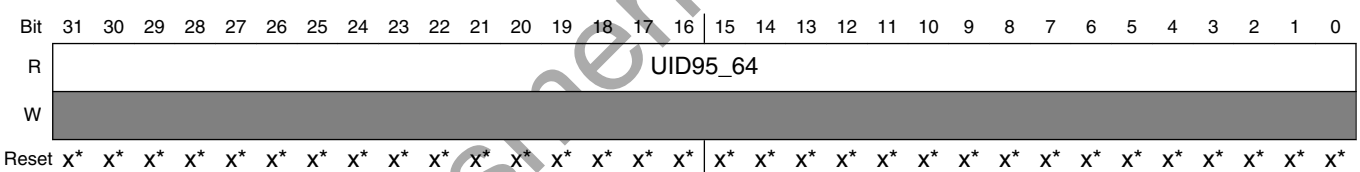
* Notes:
- x = Undefined at reset.

SIM_UIDH field descriptions

Field	Description
UID127_96	Unique Identification Unique identification for the device.

13.2.11 Unique Identification Register Mid-High (SIM_UIDMH)

Address: 0h base + 58h offset = 58h



* Notes:
- x = Undefined at reset.

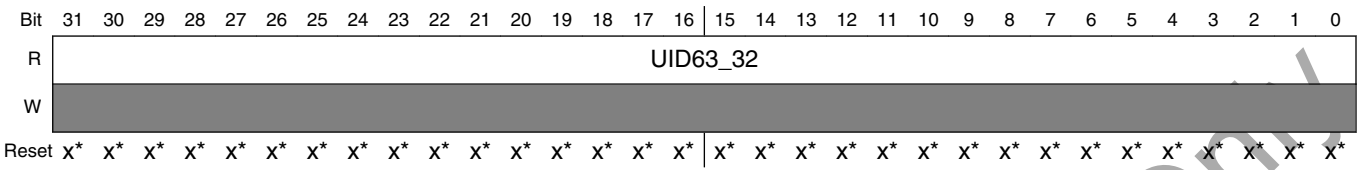
SIM_UIDMH field descriptions

Field	Description
UID95_64	Unique Identification Unique identification for the device.

For Assessment Purposes Only

13.2.12 Unique Identification Register Mid Low (SIM_UIDML)

Address: 0h base + 5Ch offset = 5Ch



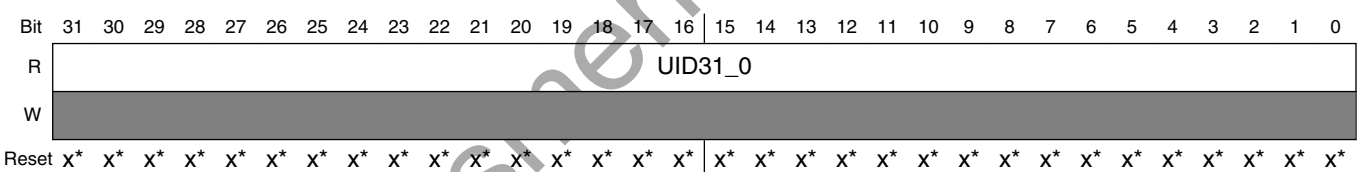
* Notes:
- x = Undefined at reset.

SIM_UIDML field descriptions

Field	Description
UID63_32	Unique Identification Unique identification for the device.

13.2.13 Unique Identification Register Low (SIM_UIDL)

Address: 0h base + 60h offset = 60h



* Notes:
- x = Undefined at reset.

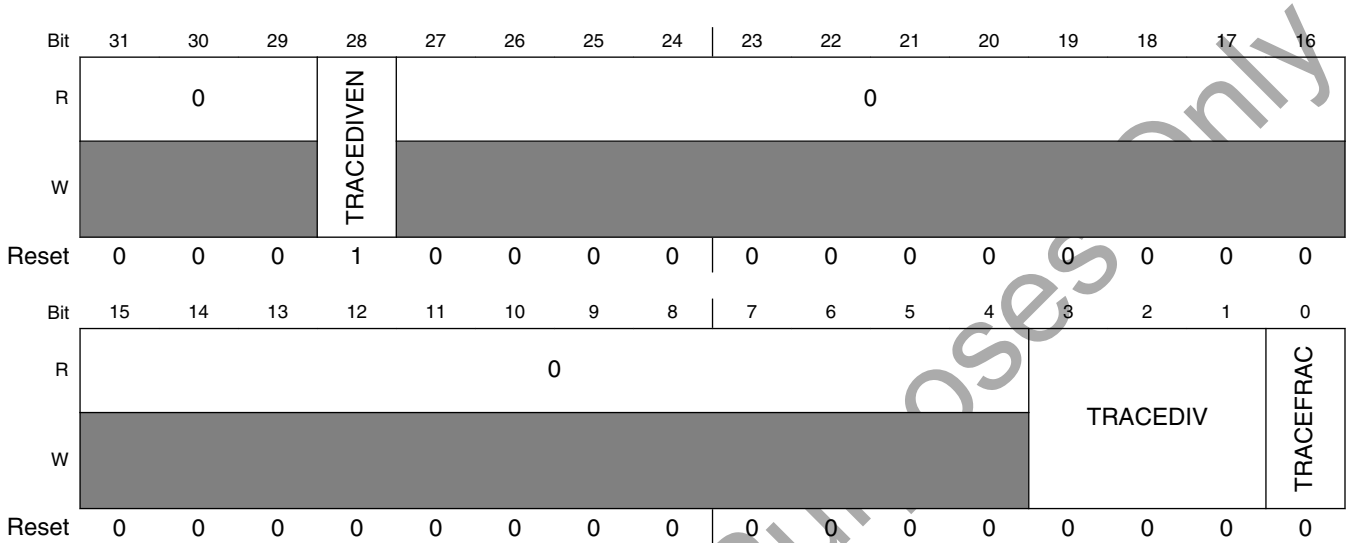
SIM_UIDL field descriptions

Field	Description
UID31_0	Unique Identification Unique identification for the device.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

13.2.14 System Clock Divider Register 4 (SIM_CLKDIV4)

Address: 0h base + 68h offset = 68h



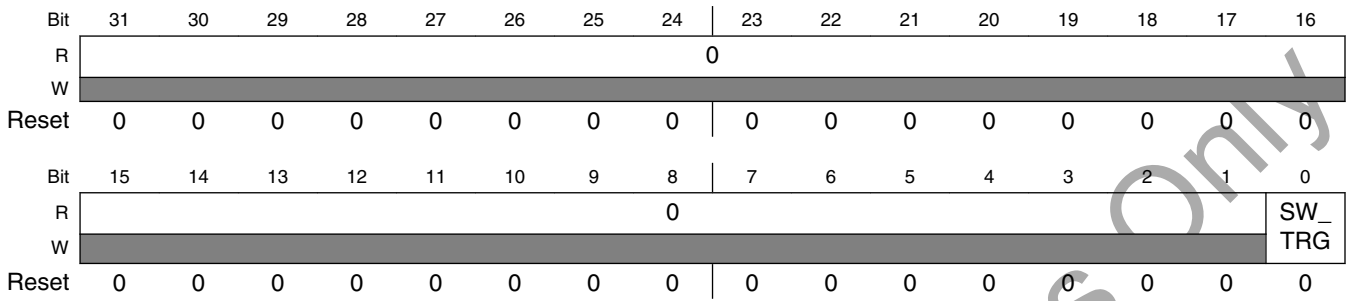
Purposes only

SIM_CLKDIV4 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 TRACEDIVEN	Debug Trace Divider Control This bit controls the Debug Trace Divider. 0 Debug trace divider disabled 1 Debug trace divider enabled
27–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–1 TRACEDIV	Trace clock divider divisor **NOTE** To configure TRACEDIV, the user must disable TRACEDIVEN at first, and then enable it after setting TRACEDIV. This field sets the divide value for the fractional clock divider used as a source for trace clock. The source clock for the trace clock is set by the SIM_CHIPCTL[TRACECLK_SEL]. Divider output clock = Divider input clock × [(TRACEFRAC+1)/(TRACEDIV+1)].
0 TRACEFRAC	Trace clock divider fraction **NOTE** To configure TRACEDIV and TRACEFRAC, the user must clear TRACEDIVEN at first to disable the trace clock divide function. This field sets the divide value for the fractional clock divider used as a source for trace clock. The source clock for the trace clock is set by the SIM_CHIPCTL[TRACECLK_SEL]. Divider output clock = Divider input clock × [(TRACEFRAC+1)/(TRACEDIV+1)].

13.2.15 Miscellaneous Control register (SIM_MISCTRL)

Address: 0h base + 6Ch offset = 6Ch



SIM_MISCTRL field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 SW_TRG	Software Trigger bit to TRGMUX

13.3 Functional description

For more information about the functions of SIM, see the Introduction section.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 14
Reset Control Module (RCM)

14.1 Chip-specific RCM information

- High-Voltage Detect Reset is not supported on this device.
- The SACKERR timing is 0.5s.

14.2 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 and ColdFire+ MCUs for further details on using the RCM.

14.3 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.

For Assessment Purposes Only

RCM memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Version ID Register (RCM_VERID)	32	R	0300_0003h	14.3.1/182
4	Parameter Register (RCM_PARAM)	32	R	See section	14.3.2/183
8	System Reset Status Register (RCM_SRS)	32	R	0000_0082h	14.3.3/185
C	Reset Pin Control register (RCM_RPC)	32	R/W	0000_0000h	14.3.4/188
18	Sticky System Reset Status Register (RCM_SSRS)	32	R/W	0000_0082h	14.3.5/190
1C	System Reset Interrupt Enable Register (RCM_SRIE)	32	R/W	0000_0000h	14.3.6/193

14.3.1 Version ID Register (RCM_VERID)

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

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

RCM_VERID field descriptions

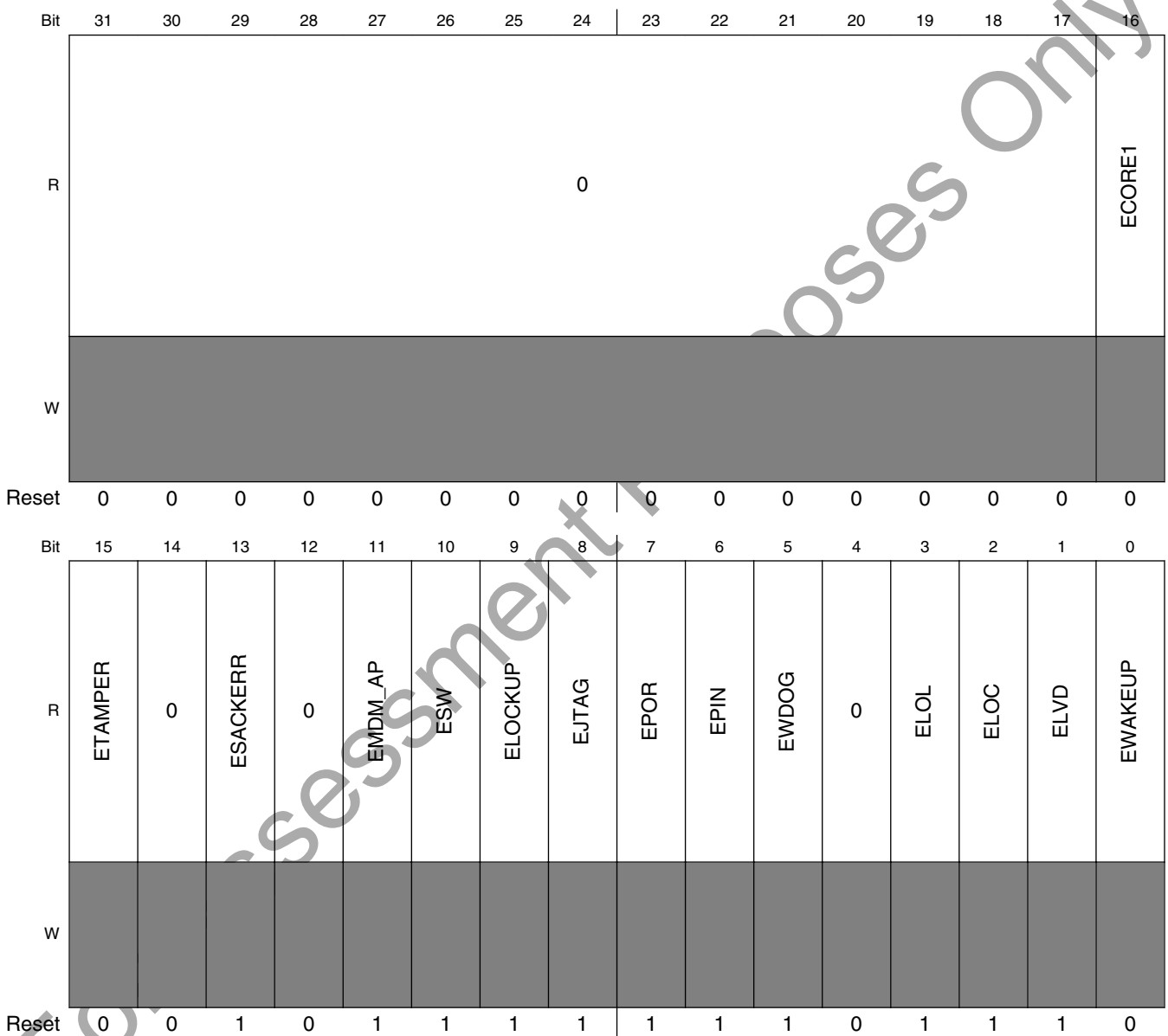
Field	Description
31–24 MAJOR	Major Version Number This read only field returns the major version number for the specification.
23–16 MINOR	Minor Version Number This read only field returns the minor version number for the specification.
FEATURE	Feature Specification Number This read only field returns the feature set number. 0x0003 Standard feature set.

For Ass... purposes O...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

14.3.2 Parameter Register (RCM_PARAM)

Address: 0h base + 4h offset = 4h



Freescale Confidential Proprietary - Non-Disclosure Agreement required

RCM_PARAM field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 ECORE1	Existence of SRS[CORE1] status indication feature

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RCM_PARAM field descriptions (continued)

Field	Description
	This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
15 ETAMPER	Existence of SRS[TAMPER] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 ESACKERR	Existence of SRS[SACKERR] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 EMDM_AP	Existence of SRS[MDM_AP] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
10 ESW	Existence of SRS[SW] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
9 ELOCKUP	Existence of SRS[LOCKUP] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
8 EJTAG	Existence of SRS[JTAG] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
7 EPOR	Existence of SRS[POR] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RCM_PARAM field descriptions (continued)

Field	Description
6 EPIN	Existence of SRS[PIN] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
5 EWDOG	Existence of SRS[WDOG] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ELOL	Existence of SRS[LOL] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
2 ELOC	Existence of SRS[LOC] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
1 ELVD	Existence of SRS[LVD] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
0 EWAKEUP	Existence of SRS[WAKEUP] status indication feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.

For A

14.3.3 System Reset Status Register (RCM_SRS)

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

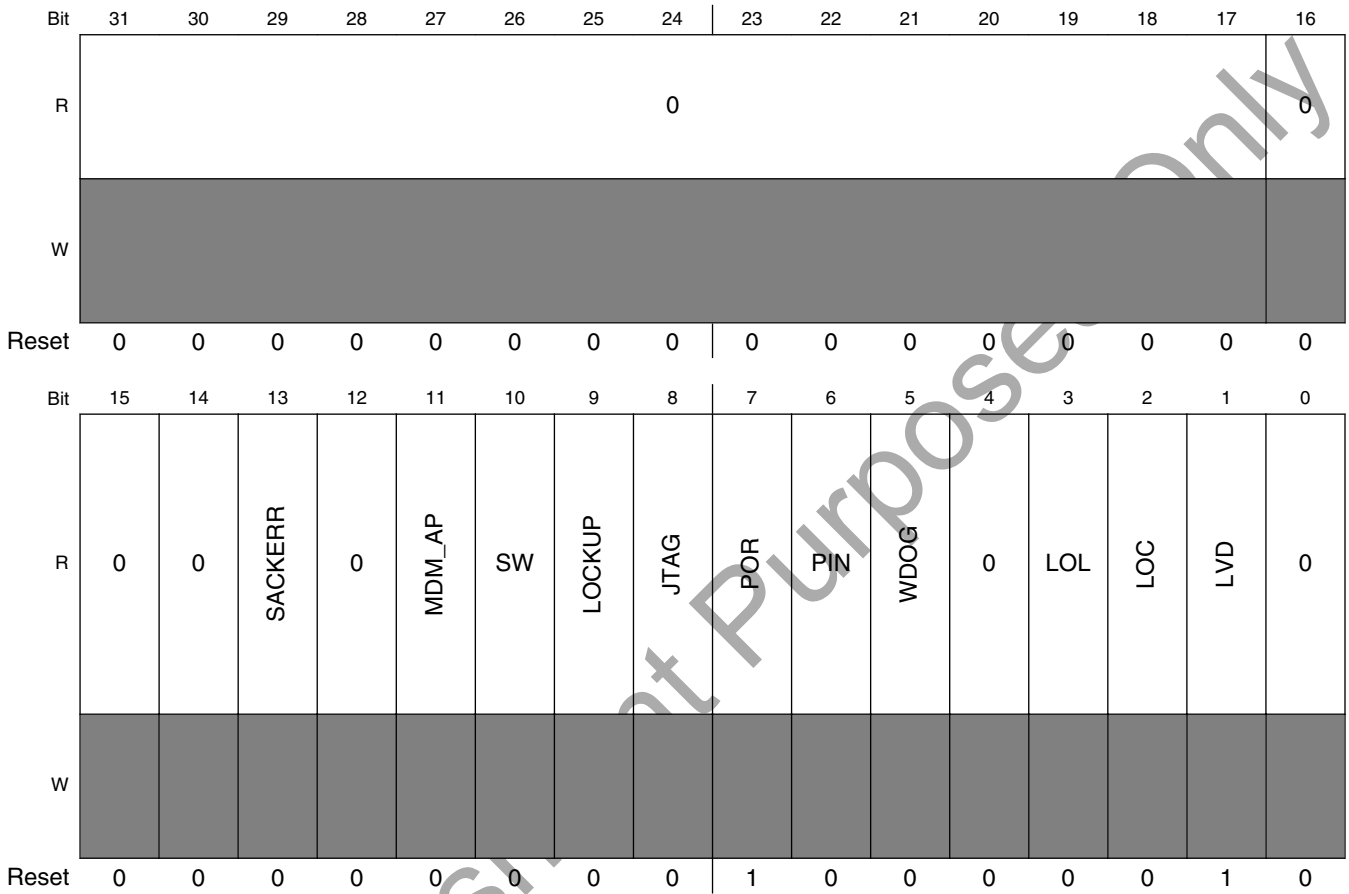
The reset value of this register depends on the reset source:

- POR (including LVD) — 0x82

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- LVD (without POR) — 0x02
- Other reset — a bit is set if its corresponding reset source caused the reset

Address: 0h base + 8h offset = 8h



RCM_SRS field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 SACKERR	Stop Acknowledge Error 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

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RCM_SRS field descriptions (continued)

Field	Description
12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 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 was not caused by host debugger system setting of the System Reset Request bit 1 Reset was caused by host debugger system setting of the System Reset Request bit
10 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
9 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
8 JTAG	JTAG generated reset Indicates a reset has been caused by the JTAG selection of certain IR codes: EXTEST, HIGHZ, and CLAMP. 0 Reset not caused by JTAG 1 Reset caused by JTAG
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 $\overline{\text{RESET}}$ (RESET_b) 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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RCM_SRS field descriptions (continued)

Field	Description
3 LOL	Loss-of-Lock Reset Indicates a reset has been caused by a loss of lock in the SCG PLL. 0 Reset not caused by a loss of lock in the PLL 1 Reset caused by a loss of lock in the PLL
2 LOC	Loss-of-Clock Reset Indicates a reset has been caused by a loss of external clock. The SCG SOSC clock monitor must be enabled for a loss of clock to be detected. Refer to the detailed SCG 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 or High-Voltage Detect Reset If PMC_LVDSC1[LVDRE] is set and the supply drops below the LVD trip voltage, an LVD reset occurs. If PMC_HVDSC1[HVDRE] is set and the supply rises above the HVD trip voltage, an HVD reset occurs. This field is also set by POR. 0 Reset not caused by LVD trip, HVD trip or POR 1 Reset caused by LVD trip, HVD trip or POR
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

14.3.4 Reset Pin Control register (RCM_RPC)

NOTE

This register is reset on Chip POR only, it is unaffected by other reset types.

NOTE

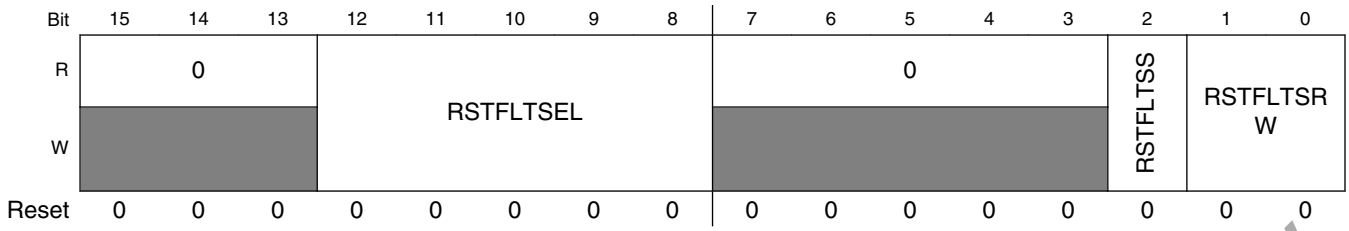
The bus clock filter is reset when disabled or when entering stop mode. The LPO filter is reset when disabled.

Address: 0h base + Ch offset = Ch

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

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required



RCM_RPC field descriptions

Field	Description
31–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 RSTFLTSEL	Reset Pin Filter Bus Clock Select Selects the reset pin bus clock filter width: • Transition lengths less than RSTFLTSEL cycles are filtered. • Transition lengths between RSTFLTSEL and (RSTFLTSEL+1) cycles (inclusive) may be filtered. • Transition lengths greater than (RSTFLTSEL+1) cycles are not filtered.
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 any stop mode. 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

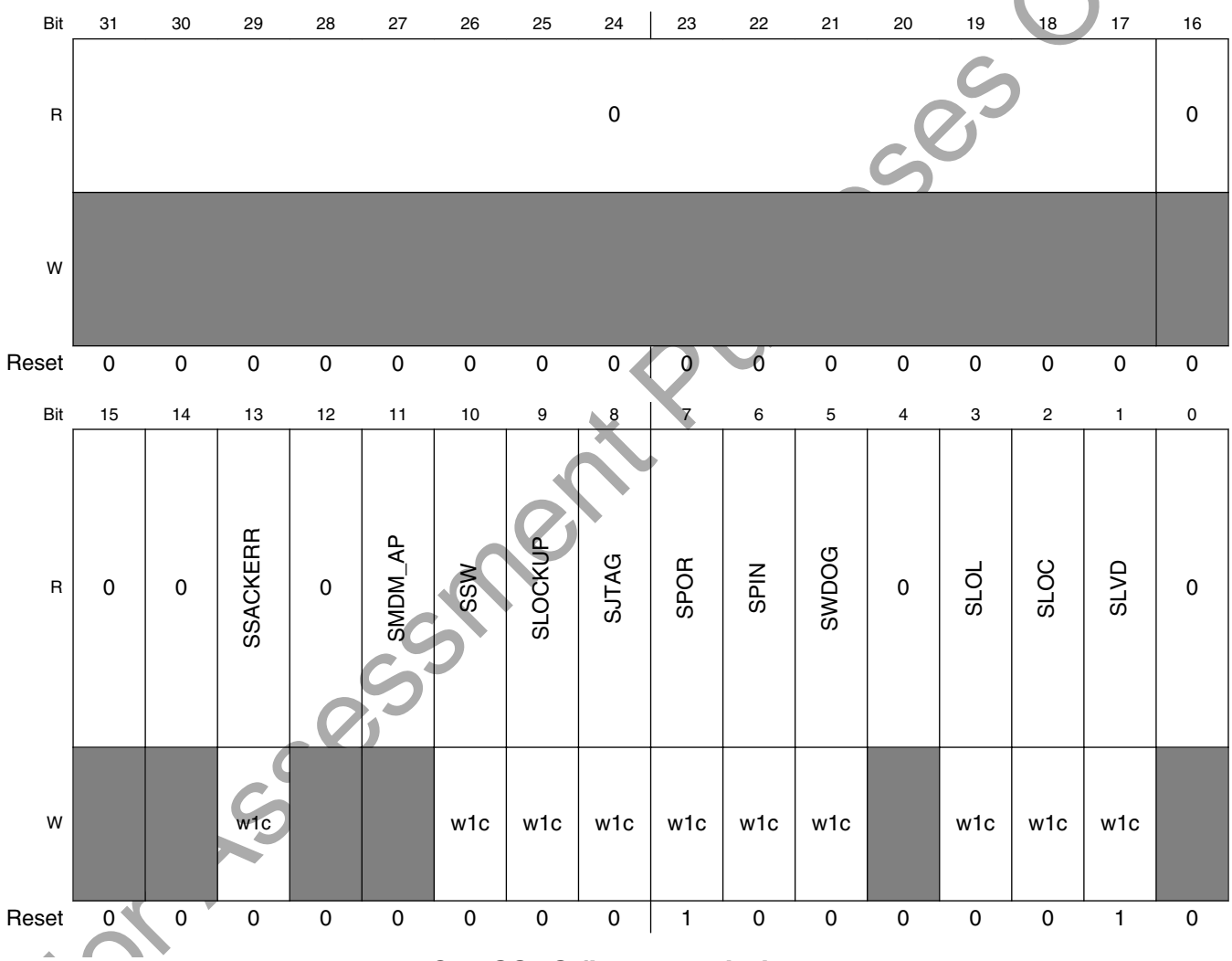
For Asse

Freescale Confidential Proprietary - Non-Disclosure Agreement required

14.3.5 Sticky System Reset Status Register (RCM_SSRS)

This register includes status flags to indicate all reset sources since the last POR or LVD Wakeup that have not been cleared by software. Software can clear the status flags by writing a logic one to a flag.

Address: 0h base + 18h offset = 18h



RCM_SSRS field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RCM_SSRS field descriptions (continued)

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 SSACKERR	Sticky Stop Acknowledge Error 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
12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 SMDM_AP	Sticky 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 was not caused by host debugger system setting of the System Reset Request bit 1 Reset was caused by host debugger system setting of the System Reset Request bit
10 SSW	Sticky 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
9 SLOCKUP	Sticky 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
8 SJTAG	Sticky JTAG generated reset Indicates a reset has been caused by the JTAG selection of certain IR codes: EXTEST, HIGHZ, and CLAMP. 0 Reset not caused by JTAG 1 Reset caused by JTAG
7 SPOR	Sticky 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 SPIN	Sticky External Reset Pin Indicates a reset has been caused by an active-low level on the external $\overline{\text{RESET}}$ (RESET_b) pin.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RCM_SSRS field descriptions (continued)

Field	Description
	0 Reset not caused by external reset pin 1 Reset caused by external reset pin
5 SWDOG	Sticky 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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 SLOL	Sticky Loss-of-Lock Reset Indicates a reset has been caused by a loss of lock in the SCG PLL. See the SCG description for information on the loss-of-lock event. 0 Reset not caused by a loss of lock in the PLL 1 Reset caused by a loss of lock in the PLL
2 SLOC	Sticky Loss-of-Clock Reset Indicates a reset has been caused by a loss of external clock. The SCG SOSC clock monitor must be enabled for a loss of clock to be detected. Refer to the detailed SCG 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 SLVD	Sticky 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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

For Ass

Freescale Confidential Proprietary - Non-Disclosure Agreement required

14.3.6 System Reset Interrupt Enable Register (RCM_SRIE)

This register delays the assertion of a system reset for a period of time (DELAY field) while an interrupt is generated. This allows software to perform a graceful shutdown. A Chip POR source cannot be delayed by this feature, and entering Stop mode terminates the delay. The SRS updates only after the system reset occurs.

Address: 0h base + 1Ch offset = 1Ch

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	SACKERR	0	MDM_AP	SW	LOCKUP	JTAG	GIE	PIN	WDOG	0	LOL	LOC	DELAY	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RCM_SRIE field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 SACKERR	Stop Acknowledge Error Interrupt 0 Interrupt disabled. 1 Interrupt enabled.
12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 MDM_AP	MDM-AP System Reset Request 0 Interrupt disabled. 1 Interrupt enabled.
10 SW	Software Interrupt 0 Interrupt disabled. 1 Interrupt enabled.

Table continues on the next page...

Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RCM_SRIE field descriptions (continued)

Field	Description
9 LOCKUP	Core Lockup Interrupt 0 Interrupt disabled. 1 Interrupt enabled.
8 JTAG	JTAG generated reset 0 Interrupt disabled. 1 Interrupt enabled.
7 GIE	Global Interrupt Enable 0 All interrupt sources disabled. 1 All interrupt sources enabled.
6 PIN	External Reset Pin Interrupt 0 Reset not caused by external reset pin 1 Reset caused by external reset pin
5 WDOG	Watchdog Interrupt 0 Interrupt disabled. 1 Interrupt enabled.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 LOL	Loss-of-Lock Interrupt 0 Interrupt disabled. 1 Interrupt enabled.
2 LOC	Loss-of-Clock Interrupt 0 Interrupt disabled. 1 Interrupt enabled.
DELAY	Reset Delay Time Configures the maximum reset delay time from when the interrupt is asserted and the system reset occurs. 00 10 LPO cycles 01 34 LPO cycles 10 130 LPO cycles 11 514 LPO cycles

For

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 15
System Mode Controller (SMC)

15.1 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.

15.2 Modes of operation

The ARM CPU has three primary modes of operation:

- Run
- Sleep
- Deep Sleep

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

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

For Assessment Purposes Only

ARM CPU mode	MCU mode
Sleep	Wait
Deep Sleep	Stop

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

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

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

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

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

ment Purposes Only

Table 15-1. 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.
HSRUN	The MCU can be run at a faster frequency compared with RUN mode and the internal supply is fully regulated. See the Power Management chapter for details about the maximum allowable frequencies.
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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

15.3 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

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	SMC Version ID Register (SMC_VERID)	32	R	0100_0000h	15.3.1/197
4	SMC Parameter Register (SMC_PARAM)	32	R	See section	15.3.2/198
8	Power Mode Protection register (SMC_PMPROT)	32	R/W	0000_0000h	15.3.3/199
C	Power Mode Control register (SMC_PMCTRL)	32	R/W	0000_0000h	15.3.4/201
10	Stop Control Register (SMC_STOPCTRL)	32	R/W	0000_0003h	15.3.5/202
14	Power Mode Status register (SMC_PMSTAT)	32	R	0000_0001h	15.3.6/204

15.3.1 SMC Version ID Register (SMC_VERID)

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

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

For Assessment + Purposes Only

SMC_VERID field descriptions

Field	Description
31–24 MAJOR	Major Version Number This read only field returns the major version number for the module specification.
23–16 MINOR	Minor Version Number This read only field returns the minor version number for the module specification.
FEATURE	Feature Specification Number This read only field returns the feature set number. 0x0000 Standard features implemented

15.3.2 SMC Parameter Register (SMC_PARAM)

Address: 0h base + 4h offset = 4h

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	EVLLS0	ELLS2	0	ELLS	0		EHSRUN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SMC_PARAM 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 0.
6 EVLLS0	Existence of VLLS0 feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
5 ELLS2	Existence of LLS2 feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ELLS	Existence of LLS feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.
2–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 EHSRUN	Existence of HSRUN feature This static bit states whether or not the feature is available on the device. 0 The feature is not available. 1 The feature is available.

For Asses

15.3.3 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.

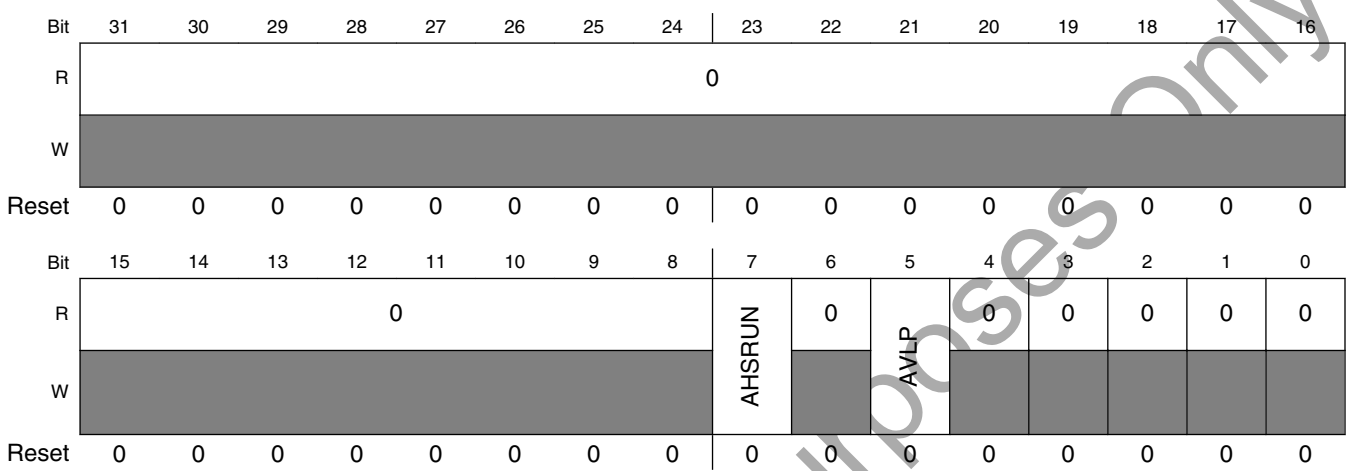
If the MCU is configured for a disallowed or reserved power mode, the MCU remains in its current power mode. For example, if the MCU is in normal RUN mode and AVLP is 0, an attempt to enter VLPR mode using PMCTRL[RUNM] is blocked and PMCTRL[RUNM] remains 00b, indicating the MCU is still in Normal Run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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

NOTE

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

Address: 0h base + 8h offset = 8h



SMC_PMPROT field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 AHSRUN	Allow High Speed Run mode Provided the appropriate control bits are set up in PMCTRL, this write-once field allows the MCU to enter High Speed Run mode (HSRUN). 0 HSRUN is not allowed 1 HSRUN is allowed
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SMC_PMPROT field descriptions (continued)

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

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

Address: 0h base + Ch offset = Ch

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								RUNM			0	STOPA	STOPM		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SMC_PMCTRL field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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. **NOTE:** RUNM may be set to HSRUN only when PMSTAT=RUN.After being programmed to HSRUN, RUNM should not be programmed back to RUN until PMSTAT=HSRUN. Also, stop mode entry should not be attempted while RUNM=HSRUN or PMSTAT=HSRUN. 00 Normal Run mode (RUN) 01 Reserved 10 Very-Low-Power Run mode (VLPR) 11 High Speed Run mode (HSRUN)
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 occured during the previous stop mode entry sequence, preventing the system from entering that mode. This field is cleared by reset or 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 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 Reserved 101 Reserved 110 Reseved 111 Reserved

15.3.5 Stop Control Register (SMC_STOPCTRL)

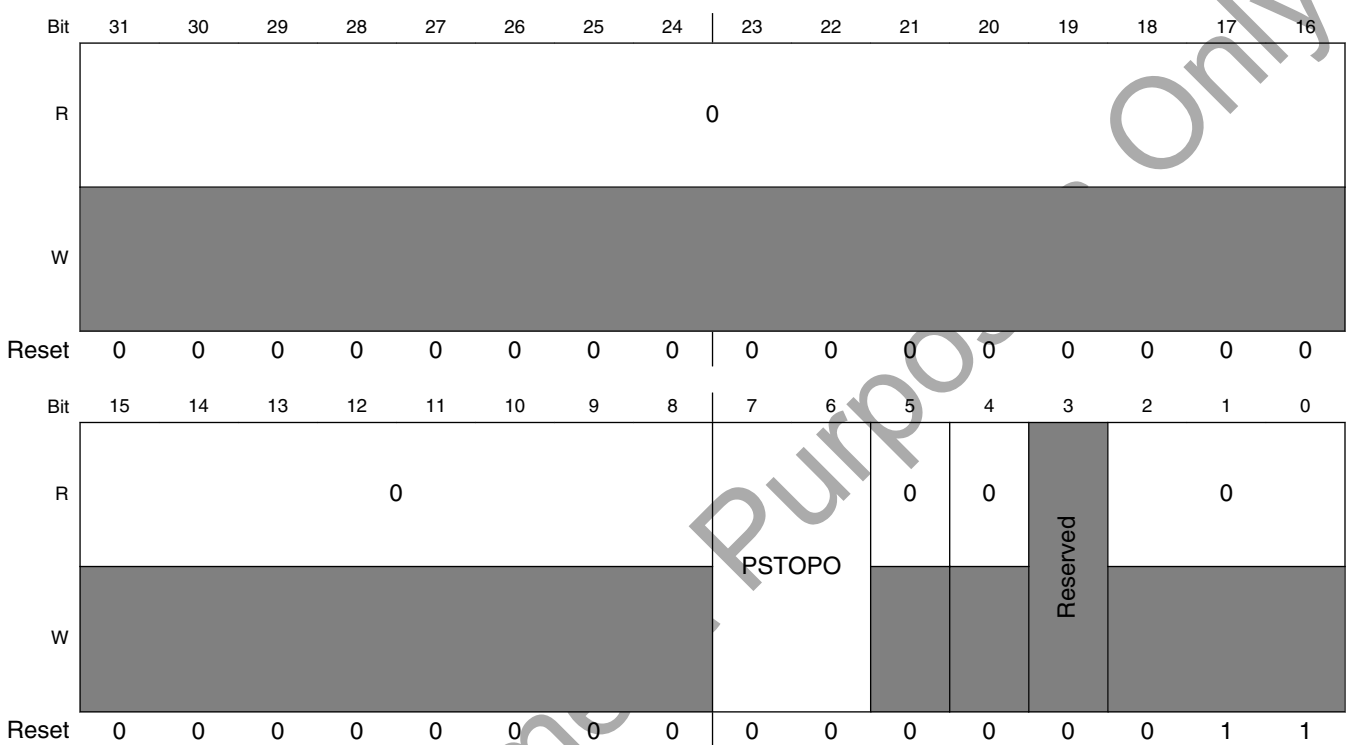
The STOPCTRL register provides various control bits allowing the user to fine tune power consumption during the stop mode selected by the STOPM field.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

NOTE

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

Address: 0h base + 10h offset = 10h



SMC_STOPCTRL 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 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 mode, the PMC, SCG 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 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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SMC_STOPCTRL field descriptions (continued)

Field	Description
3 Reserved	This field is reserved. This bit is reserved for future expansion and should always be written zero.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

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

Address: 0h base + 14h offset = 14h

Bit	31	30	29	28	27	26	25	24	23	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																								PMSTAT							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SMC_PMSTAT field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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 Reserved 0100_0000 Reserved 1000_0000 Current power mode is HSRUN

ment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

15.4 Functional description

15.4.1 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.

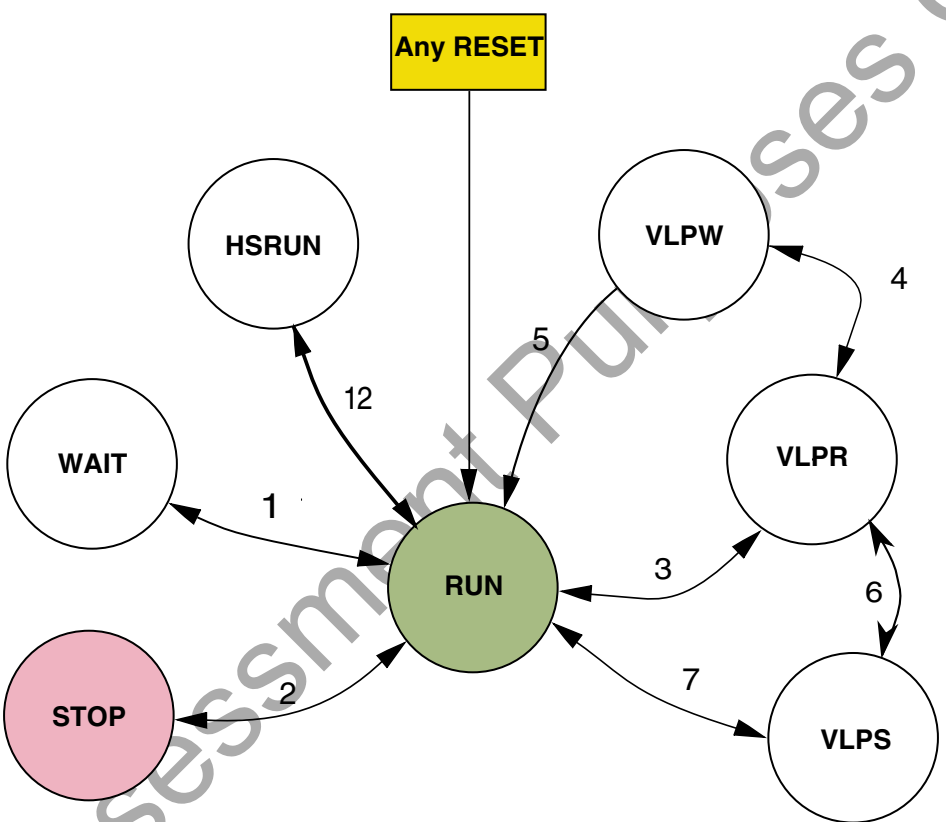


Figure 15-1. Power mode state diagram

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

Table 15-2. 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]

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 15-2. Power mode transition triggers (continued)

Transition #	From	To	Trigger conditions
			Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	STOP	RUN	Interrupt or Reset
3	RUN	VLPR	The core, system, bus and flash clock frequencies and SCG clocking mode are restricted in this mode. See the Power Management chapter for the maximum allowable frequencies and SCG 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
12	RUN	HSRUN	Set PMPROT[AHSRUN]=1, PMCTRL[RUNM]=11.
	HSRUN	RUN	Set PMCTRL[RUNM]=00 or Reset

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

15.4.2 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.

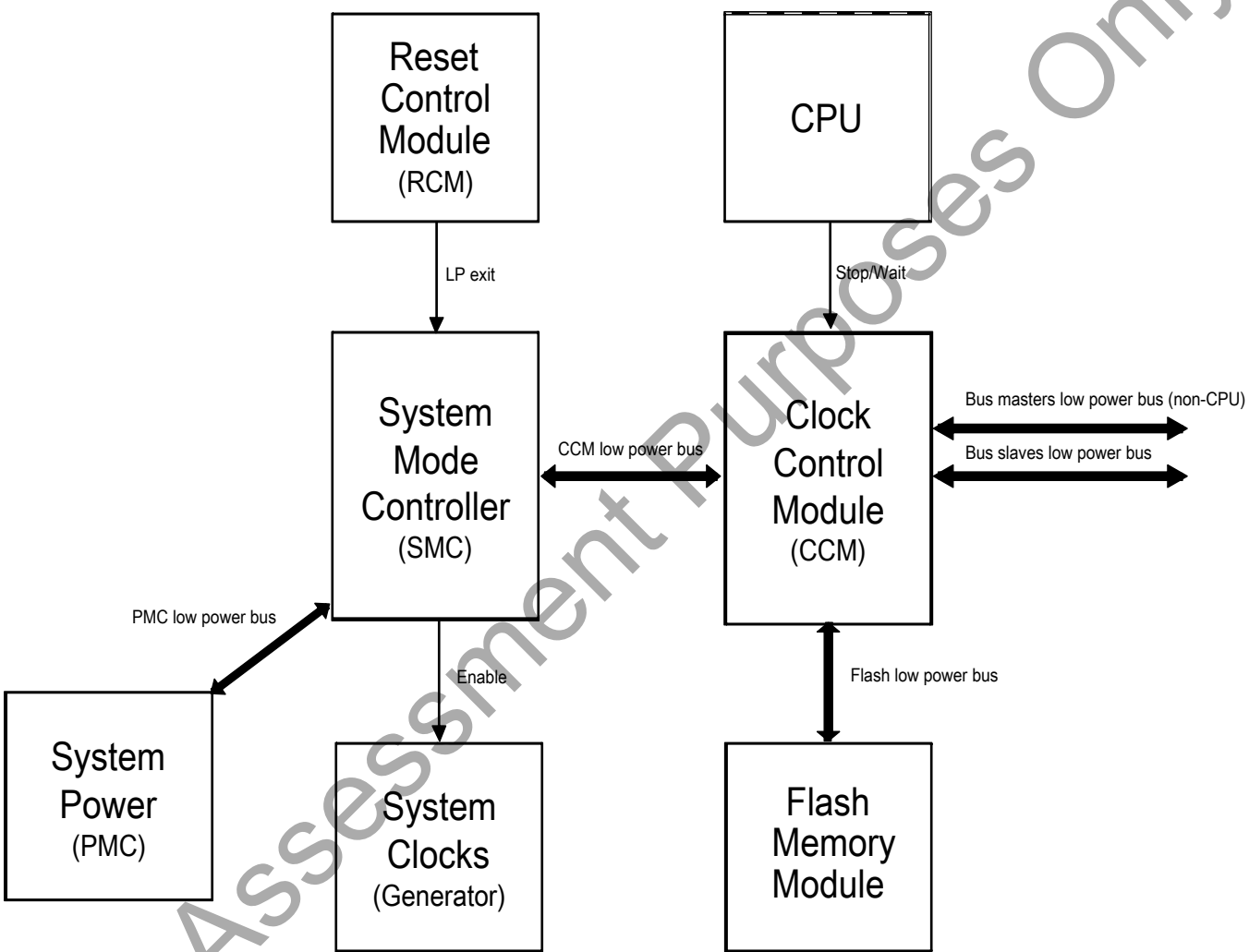


Figure 15-2. Low-power system components and connections

For Assessment Purposes Only

15.4.2.1 Stop mode entry sequence

Entry into a low-power stop mode (Stop, VLPS) 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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 SCG.
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.

15.4.2.2 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 SCG.
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.

15.4.2.3 Aborted stop mode entry

If an interrupt 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 interrupt occurs before the PMC begins the transition to stop mode regulation. After this point, the interrupt 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.

15.4.2.4 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

15.4.2.5 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.

15.4.3 Run modes

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

- Run (RUN)
- Very Low-Power Run (VLPR)
- High Speed Run (HSRUN)

15.4.3.1 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.

15.4.3.2 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 PCC's registers.

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

- All clock monitors in the SCG 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.

For Assessment Purposes Only

- 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 SCG module or any clock divider registers. Module clock enables in the PCC 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.

15.4.3.3 High Speed Run (HSRUN) mode

In HSRUN mode, the on-chip voltage regulator remains in a run regulation state, but with a slightly elevated voltage output. In this state, the MCU is able to operate at a faster frequency compared to normal RUN mode. For the maximum allowable frequencies, see the Power Management chapter.

While in this mode, the following restrictions must be adhered to:

- The maximum allowable change in frequency of the system, bus, flash or core clocks is restricted to 2x (double the frequency).
- Stop mode entry is not supported from HSRUN.
- Modifications to clock gating control bits are prohibited.
- Flash programming/erasing is not allowed.

To enter HSRUN mode, set PMPORT[AHSRUN]=HSRUN and set PMCTRL[RUNM]=HSRUN. Before increasing clock frequencies, the PMSTAT register should be polled to determine when the system has completed entry into HSRUN mode. To reenter normal RUN mode, clear RUNM. Any reset will also clear RUNM and cause the system to exit to normal RUN mode after the MCU exits its reset flow.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

15.4.4 Wait modes

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

- Wait
- Very-Low Power Wait (VLPW)

15.4.4.1 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.

15.4.4.2 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.

15.4.5 Stop modes

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

The stop modes range from:

For Assessment Purposes Only

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

to:

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

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

NOTE

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

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)

15.4.5.1 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 SCG 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.

15.4.5.2 Very-Low-Power Stop (VLPS) mode

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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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.

15.4.6 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 SCG-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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 16
Power Management Controller (PMC)

16.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances, see the chip configuration information.

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

16.2 Features

The PMC features include:

- Internal voltage regulator offering a variety of power modes
- Active POR providing brown-out detect
- Low voltage reset (LVR)
- Low voltage detect supporting two low voltage trip points and interrupt
- Low power oscillator (LPO) with a typical frequency of 128 kHz

16.3 Modes of Operation

16.3.1 Full Performance Mode (FPM)

For the following Chip Power Modes, the internal voltage regulator is in full performance mode: HSRUN, RUN, WAIT.

For Assessment Purposes Only

16.3.2 Low Power Mode (LPM)

For the following Chip Power Modes, the internal voltage regulator is in low power mode: STOP, VLPR, VLPW, VLPS.

16.4 Low Voltage Detect (LVD) System

NOTE

The low voltage detect system (Low voltage detect flag, Low voltage warning flag and Low voltage detect reset generation) is disabled in low power mode.

This device includes a system to guard against low voltage conditions. This protects memory contents and controls MCU system states during supply voltage variations. The system is comprised of a power-on reset (POR) circuit and a LVD circuit with two trip points. The LVD is disabled upon entering low power mode.

Two flags are available to indicate the status of the low voltage detect system:

- The low voltage detect flag (LVDF) operates in a level sensitive manner. The LVDF bit is set when the supply voltage falls below the trip point (V_{LVD}). The LVDF bit is cleared by writing one to the LVDACK bit, but only if the internal supply has returned above the trip point; otherwise, the LVDF bit remains set.
- The low voltage warning flag (LVWF) operates in a level sensitive manner. The LVWF bit is set when the supply voltage falls below the selected monitor trip point (V_{LVW}). The LVWF bit is cleared by writing one to the LVWACK bit, but only if the internal supply has returned above the trip point; otherwise, the LVWF bit remains set.

16.4.1 Low Voltage Reset (LVR) Operation

If the supply voltage falls below the reset trip point (V_{LVR}), a system reset will be generated.

If PMC_LVDSC1[LVDRE] is set and the supply voltage falls below V_{LVD}, a system reset will be generated.

PMC_LVDSC1[LVDF] will be cleared by system reset, so after recovery PMC_LVDSC1[LVDF] will read zero. Usage of PMC_LVDSC1[LVDF] is intended for LVD interrupt opteration only (for example, PMC_LVDSC1[LVDIE] = 1 and PMC_LVDSC1[LVDRE] = 0).

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

16.4.2 LVD Interrupt Operation

By configuring the LVD circuit for interrupt operation (LVDIE set), PMC_LVDSC1[LVDF] is set and an LVD interrupt request occurs upon detection of a low voltage condition. The LVDF bit is cleared by writing one to the PMC_LVDSC1[LVDACK] bit, when the supply returns to above the trip point.

16.4.3 Low-voltage warning (LVW) interrupt operation

The LVD system contains a low-voltage warning flag (LVWF) 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 the PMC_LVDSC2[LVWIE] bit. If enabled, an LVW interrupt request occurs when the LVWF is set. LVWF is cleared by writing one to the PMC_LVDSC2[LVWACK] bit, when the supply returns to above the trip point.

16.5 Memory Map and Register Definition

This sections provides the detailed information of all registers for the PMC module.

NOTE

Different portions of PMC registers are reset only by particular reset types. Each register's description provides details.

NOTE

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

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Low Voltage Detect Status and Control 1 Register (PMC_LVDSC1)	8	R/W	See section	16.5.1/218
1	Low Voltage Detect Status and Control 2 Register (PMC_LVDSC2)	8	R/W	00h	16.5.2/219
2	Regulator Status and Control Register (PMC_REGSC)	8	R/W	See section	16.5.3/220
3	Low Voltage Reset Flags Register (PMC_LVRFLG)	8	R/W	11_1111h	16.5.4/221
4	Low Power Oscillator Trim Register (PMC_LPOTRIM)	8	R/W	See section	16.5.5/222

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

16.5.1 Low Voltage Detect Status and Control 1 Register (PMC_LVDSC1)

This register contains status and control bits to support the low voltage detect function.

NOTE

When the internal voltage regulator is in low power mode, the LVD system is disabled, regardless of the PMC_LVDSC1 settings.

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

Bit	7	6	5	4	3	2	1	0
Read	LVDF		LVDIE	LVDRE	0			
Write		LVDACK						
Reset	0	0	0	u*	0	0	0	0
POR	0	0	0	0	0	0	0	0

* Notes:
- u = Unaffected by reset.

Purposes Only

PMC_LVDSC1 field descriptions

Field	Description
7 LVDF	Low Voltage Detect Flag This bit's read-only status bit indicates a low-voltage detect event. The threshold voltage is V_{LVD}. 0 Low-voltage event not detected 1 Low-voltage event detected
6 LVDACK	Low Voltage Detect Acknowledge This write-only bit is used to acknowledge low voltage detection errors. Write 1 to clear LVDF. Read always return 0.
5 LVDIE	Low Voltage Detect Interrupt Enable This bit 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 bit enables the low voltage detect events to generate a system reset. 0 No system resets on low voltage detect events. 1 If the supply voltage falls below V_{LVD}, a system reset will be generated.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

16.5.2 Low Voltage Detect Status and Control 2 Register (PMC_LVDSC2)

This register contains status and control bits to support the low voltage warning (LVW) function.

NOTE

When the internal voltage regulator is in low power mode, the LVD system is disabled regardless of the PMC_LVDSC2 settings.

Address: 0h base + 1h offset = 1h

Bit	7	6	5	4	3	2	1	0
Read	LVWF		LVWIE	0				
Write		LVWACK						
Reset	0	0	0	0	0	0	0	0

PMC_LVDSC2 field descriptions

Field	Description
7 LVWF	Low-Voltage Warning Flag This bit read-only status bit indicates a low-voltage detect event. The threshold voltage is V_{LVW}. 0 Low-voltage warning event not detected 1 Low-voltage warning event detected
6 LVWACK	Low-Voltage Warning Acknowledge This write-only bit is used to acknowledge low voltage warning errors. Write 1 to clear LVWF. Reads always return 0.
5 LVWIE	Low-Voltage Warning Interrupt Enable This bit enables hardware interrupt requests for LVWF. 0 Hardware interrupt disabled (use polling) 1 Request a hardware interrupt when LVWF=1
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

For Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

16.5.3 Regulator Status and Control Register (PMC_REGSC)

This register contains general control and status bits for the regulator and the LPO.

Address: 0h base + 2h offset = 2h

Bit	7	6	5	4	3	2	1	0
Read	LPODIS	LPOSTAT	0			REGFPM	CLKBIASDIS	BIASEN
Write								
Reset	u*	*	0	0	0	1	0	0
POR	0	*	0	0	0	1	0	0

* Notes:
- u = Unaffected by reset.
- LPOSTAT field: Reset value is undefined.

ses Only

PMC_REGSC field descriptions

Field	Description
7 LPODIS	LPO Disable Bit This bit enables or disable the low power oscillator. **NOTE:** After disabling the LPO a time of 2 LPO clock cycles is required before it is allowed to enable it again. Violating this waiting time of 2 cycles can result in malfunction of the LPO. 0 Low power oscillator enabled 1 Low power oscillator disabled
6 LPOSTAT	LPO Status Bit This bit shows the status of the LPO clock to be either in high phase (logic 1) or low phase (logic 0) of the clock period. Software can poll this status bit to measure actual LPO clock frequency and eventually use the LPOTRIM[4:0] register to change the LPO frequency. 0 Low power oscillator in low phase 1 Low power oscillator in high phase
5–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 REGFPM	Regulator in Full Performance Mode Status Bit This read-only bit provides the current status of the internal voltage regulator. 0 Regulator is in low power mode or transition to/from 1 Regulator is in full performance mode
1 CLKBIASDIS	Clock Bias Disable Bit This bit disables the bias currents and reference voltages for some clock modules in order to further reduce power consumption in STOP or VLPS mode if all clocks are disabled. The bias currents and reference voltages for LPFLL (if available on device) are always disabled in RPM. **Note: Using this bit it must be ensured that respective clock modules are disabled in STOP or VLPS mode. Else severe malfunction of clock modules will happen.**

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PMC_REGSC field descriptions (continued)

Field	Description
	0 No effect 1 In STOP or VLPS mode the bias currents and reference voltages for the following clock modules are disabled: SIRC, FIRC, PLL. (if available on device)
0 BIASEN	Bias Enable Bit This bit enables source and well biasing for the core logic in low power mode. In full performance mode this bit has no effect. This is useful to further reduce MCU power consumption in low power mode. 0 Biasing disabled, core logic can run in full performance 1 Biasing enabled, core logic is slower and there are restrictions in allowed system clock speed (see *Data Sheet* for details)

16.5.4 Low Voltage Reset Flags Register (PMC_LVRFLG)

This register shows with sticky flags on which power domains a low voltage reset has occurred.

Address: 0h base + 3h offset = 3h

Bit	7	6	5	4	3	2	1	0
Read	0		LVR3FLSF	LVR3F	LVRXLPF	LVRXF	LVRLPF	LVRF
Write			w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	1	0	0	0	1

PMC_LVRFLG field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 LVR3FLSF	LVR 3 V Flash memory Flag Displays that a low voltage reset event has occurred on the internal 3V power domain of flash memory. 0 Low voltage reset event has not occurred on internal 3 V power domain of flash memory. 1 Low voltage reset event has occurred on internal 3 V power domain of flash memory.
4 LVR3F	LVR 3 V Flag Displays that a low voltage reset event has occurred on the internal 3V power domain. 0 Low voltage reset event has not occurred on internal 3 V power domain. 1 Low voltage reset event has occurred on internal 3 V power domain.
3 LVRXLPF	LVR external in low power mode Flag Displays that a low voltage reset event has occurred on the external 5V power domain while in low power mode. 0 Low voltage reset event has not occurred on external 5 V power domain while in low power mode. 1 Low voltage reset event has occurred on external 5 V power domain while in low power mode.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PMC_LVRFLG field descriptions (continued)

Field	Description
2 LVRXF	LVR External Flag Displays that a low voltage reset even has occurred on the external 5 V power domain. 0 Low voltage reset event has not occurred on external 5 V power domain. 1 Low voltage reset event has occurred on external 5 V power domain.
1 LVRLPF	LVR in low power mode core Flag Displays that a low voltage reset event has occurred on the chip internal core logic power domain while in low power mode. 0 Low voltage reset event has not occurred. 1 Low voltage reset event has occurred.
0 LVRF	LVR Core Flag Displays that a low voltage reset event has occurred on the chip internal core logic power domain. 0 Low voltage reset has not occurred. 1 Low voltage reset has occurred.

16.5.5 Low Power Oscillator Trim Register (PMC_LPOTRIM)

This register contains the period trimming bits for the low power oscillator.

Table 16-1. Trimming effect of LPOTRIM[4:0]

LPOTRIM[4:0]	Decimal	Period of LPO clock
10000	–16	lowest
10001	–15	increasing
...	...	
11110	–2	
11111	–1	
00000	0	typical 128 kHz
00001	+1	increasing
...	...	
01110	+14	
01111	+15	highest

NOTE

The LPO trimming bits represent signed values. Starting from -16 the period of the LPO clock will increase monotonically (for example, frequency decreases monotonically).

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 4h offset = 4h

Bit	7	6	5	4	3	2	1	0
Read	0			LPOTRIM				
Write								
Reset	0	0	0	*	*	*	*	*
POR	0	0	0	0*	0*	0*	0*	0*

* Notes:
- LPOTRIM field: After POR reset, automatically loaded from Flash Memory IFR after Reset (normal system reset).

PMC_LPOTRIM field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LPOTRIM	LPO trimming bits These bits are used for trimming the frequency of the low power oscillator. See the table above for trimming effect.

For Assessment Purpose Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 17
Miscellaneous Control Module (MCM)

17.1 Introduction

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

17.1.1 Features

The MCM includes the following features:

- Program-visible information on the platform configuration and revision

17.2 Memory map/register descriptions

The memory map and register descriptions below describe the registers using byte addresses.

MCM memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
8	Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)	16	R	0007h	17.2.1/226
A	Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)	16	R	0007h	17.2.2/227
C	Core Platform Control Register (MCM_CPCR)	32	R/W	0000_0000h	17.2.3/227
10	Interrupt Status and Control Register (MCM_ISCR)	32	R	0000_0000h	17.2.4/229
20	Store Buffer Fault address register (MCM_FADR)	32	R	Undefined	17.2.5/232
24	Store Buffer Fault Attributes register (MCM_FATR)	32	R	Undefined	17.2.6/232
28	Store Buffer Fault Data Register (MCM_FDR)	32	R	Undefined	17.2.7/234
30	Process ID register (MCM_PID)	32	R/W	0000_0000h	17.2.8/235

Table continues on the next page...

Assessment Purposes Only

MCM memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40	Compute Operation Control Register (MCM_CPO)	32	R/W	0000_0000h	17.2.9/236
400	Local Memory Descriptor Register (MCM_LMDR0)	32	R/W	See section	17.2.10/237
404	Local Memory Descriptor Register (MCM_LMDR1)	32	R/W	See section	17.2.10/237
408	Local Memory Descriptor Register (MCM_LMDR2)	32	R/W	See section	17.2.10/237
480	LMEM Parity & ECC Control Register (MCM_LMPECR)	32	R/W	0000_0000h	17.2.11/241
488	LMEM Parity & ECC Interrupt Register (MCM_LMPEIR)	32	R/W	0000_0000h	17.2.12/242
490	LMEM Fault Address Register (MCM_LMFAR)	32	R	0000_0000h	17.2.13/243
494	LMEM Fault Attribute Register (MCM_LMFATR)	32	R/W	0000_0000h	17.2.14/244
4A0	LMEM Fault Data High Register (MCM_LMFDHR)	32	R	0000_0000h	17.2.15/245
4A4	LMEM Fault Data Low Register (MCM_LMFDLR)	32	R	0000_0000h	17.2.16/245

17.2.1 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: 0h base + 8h offset = 8h

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Freescale Confidential Proprietary - Non-Disclosure Agreement required

17.2.2 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: 0h base + Ah offset = Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								AMC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

MCM_PLAMC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
AMC	Each bit in the AMC field indicates whether there is a corresponding connection to the AXBS master input port. 0 A bus master connection to AXBS input port *n* is absent 1 A bus master connection to AXBS input port *n* is present

17.2.3 Core Platform Control Register (MCM_CPCR)

Control Register defines the arbitration and protection schemes for the two system RAM arrays.

NOTE

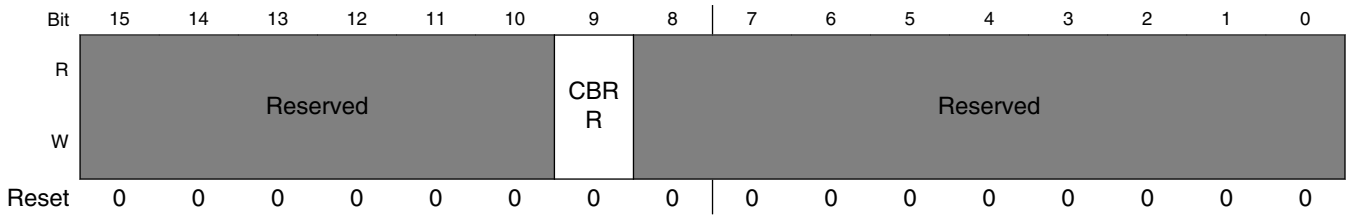
Bits 23-0 are undefined after reset.

Address: 0h base + Ch offset = Ch

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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required



MCM_CPCR field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 SRAMLWP	SRAM_L Write Protect When this bit is set, writes to SRAM_L array generates a bus error.
29–28 SRAMLAP	SRAM_L arbitration priority Defines the arbitration scheme and priority for the processor and SRAM backdoor accesses to the SRAM_L array. 00 Round robin 01 Special round robin (favors SRAM backdoor accesses over the processor) 10 Fixed priority. Processor has highest, backdoor has lowest 11 Fixed priority. Backdoor has highest, processor has lowest
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 SRAMUWP	SRAM_U write protect When this bit is set, writes to SRAM_U array generates a bus error.
25–24 SRAMUAP	SRAM_U arbitration priority Defines the arbitration scheme and priority for the processor and SRAM backdoor accesses to the SRAM_U array. 00 Round robin 01 Special round robin (favors SRAM backdoor accesses over the processor) 10 Fixed priority. Processor has highest, backdoor has lowest 11 Fixed priority. Backdoor has highest, processor has lowest
23–10 Reserved	This field is reserved.
9 CBRR	Crossbar round-robin arbitration enable Configures the crossbar slave ports to fixed-priority or round-robin arbitration. 0 Fixed-priority arbitration 1 Round-robin arbitration
Reserved	This field is reserved.

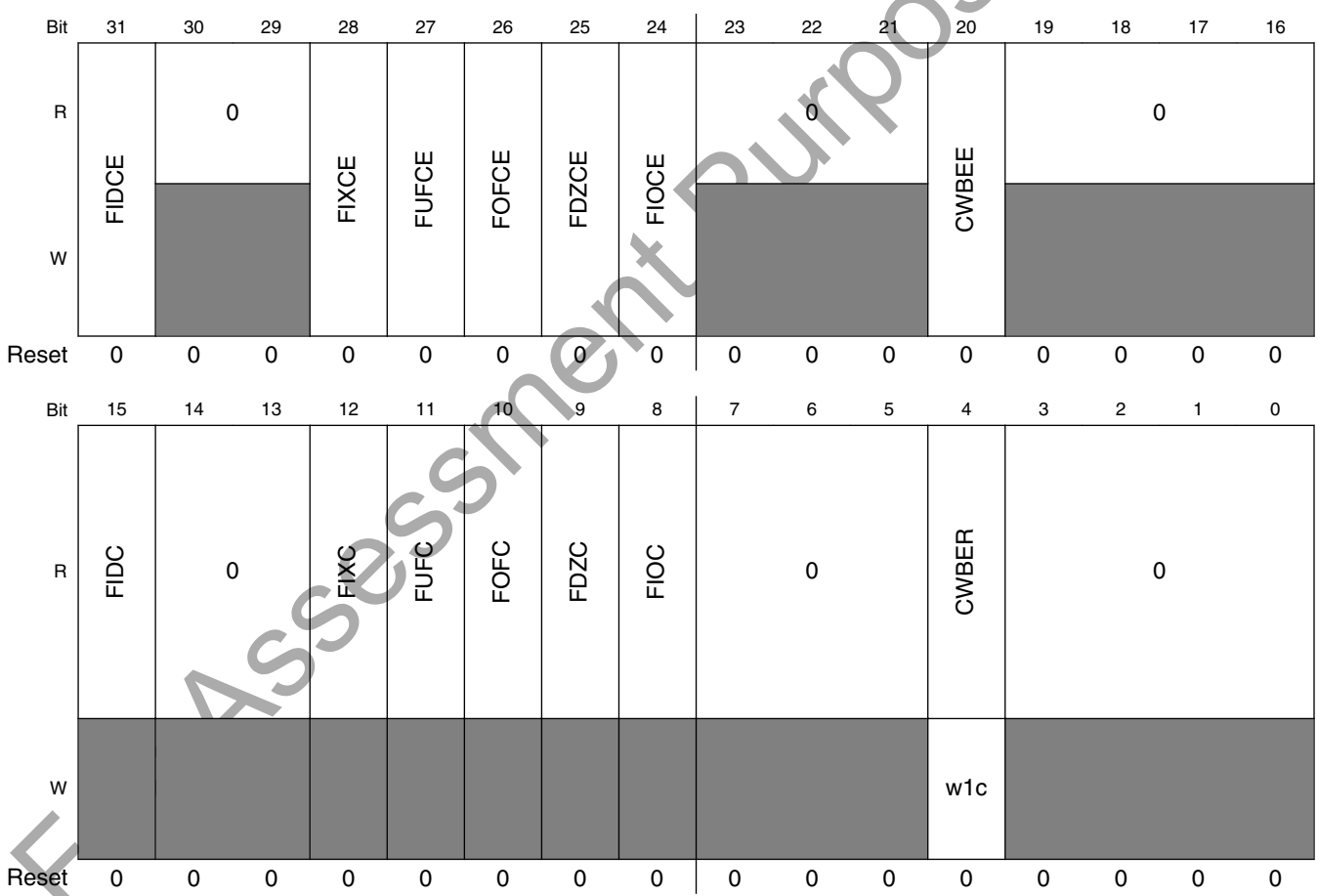
Freescale Confidential Proprietary - Non-Disclosure Agreement required

17.2.4 Interrupt Status and Control Register (MCM_ISCR)

The MCM_ISCR register defines the configuration and reports status for a number of core-related interrupt exception conditions. It includes the enable and status bits associated with the core's floating-point exceptions, and bus errors associated with the core's cache write buffer. The individual event indicators are first qualified with their exception enables and then logically summed to form an interrupt request sent to the core's NVIC.

Bits 15-8 are read-only indicator flags based on the processor's FPSCR register. Attempted writes to these bits are ignored. Once set, the flags remain asserted until software clears the corresponding FPSCR bit.

Address: 0h base + 10h offset = 10h



MCM_ISCR field descriptions

Field	Description
31 FIDCE	FPU input denormal interrupt enable

Table continues on the next page...

Freescale Assessment Purposes Only

MCM_ISCR field descriptions (continued)

Field	Description
	0 Disable interrupt 1 Enable interrupt
30–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 FIXCE	FPU inexact interrupt enable 0 Disable interrupt 1 Enable interrupt
27 FUFCE	FPU underflow interrupt enable 0 Disable interrupt 1 Enable interrupt
26 FOFCE	FPU overflow interrupt enable 0 Disable interrupt 1 Enable interrupt
25 FDZCE	FPU divide-by-zero interrupt enable 0 Disable interrupt 1 Enable interrupt
24 FIOCE	FPU invalid operation interrupt enable 0 Disable interrupt 1 Enable interrupt
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 CWBEE	Cache write buffer error enable Enables the generation of an interrupt in response to a bus error termination reported on a system bus transfer initiated from the cache's write buffer. 0 Disable error interrupt 1 Enable error interrupt
19–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 FIDC	FPU input denormal interrupt status This read-only bit is a copy of the core's FPSCR[IDC] bit and signals input denormalized number has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IDC] bit. 0 No interrupt 1 Interrupt occurred
14–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 FIXC	FPU inexact interrupt status This read-only bit is a copy of the core's FPSCR[IXC] bit and signals an inexact number has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IXC] bit.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MCM_ISCR field descriptions (continued)

Field	Description
	0 No interrupt 1 Interrupt occurred
11 FUFC	FPU underflow interrupt status This read-only bit is a copy of the core's FPSCR[UFC] bit and signals an underflow has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[UFC] bit. 0 No interrupt 1 Interrupt occurred
10 FOFC	FPU overflow interrupt status This read-only bit is a copy of the core's FPSCR[OFC] bit and signals an overflow has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[OFC] bit. 0 No interrupt 1 Interrupt occurred
9 FDZC	FPU divide-by-zero interrupt status This read-only bit is a copy of the core's FPSCR[DZC] bit and signals a divide by zero has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[DZC] bit. 0 No interrupt 1 Interrupt occurred
8 FIOC	FPU invalid operation interrupt status This read-only bit is a copy of the core's FPSCR[IOC] bit and signals an illegal operation has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IOC] bit. 0 No interrupt 1 Interrupt occurred
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 CWBER	Cache write buffer error status Signals a data transfer from the core's cache write buffer was terminated with a bus error. This bit only sets when the corresponding enable bit (CWBEE) is set. The corresponding core fault address, attributes and write data are typically retrieved from the FADR, FATR, and FDR registers during the interrupt service routine before clearing the CWBER flag. 0 No error 1 Error occurred
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

17.2.5 Store Buffer Fault address register (MCM_FADR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting address is captured in the MCM_FADR register. The MCM logic supports capturing a single cache write buffer bus error event; if a subsequent error is detected before the captured error information has been read from the corresponding registers and the MCM_ISCR[CWBER] indicator cleared, the MCM_FATR[BEOVR] flag is set. However, no additional information is captured.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. Attempted writes have no effect.

Address: 0h base + 20h offset = 20h

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

MCM_FADR field descriptions

Field	Description
ADDRESS	Fault address

17.2.6 Store Buffer Fault Attributes register (MCM_FATR)

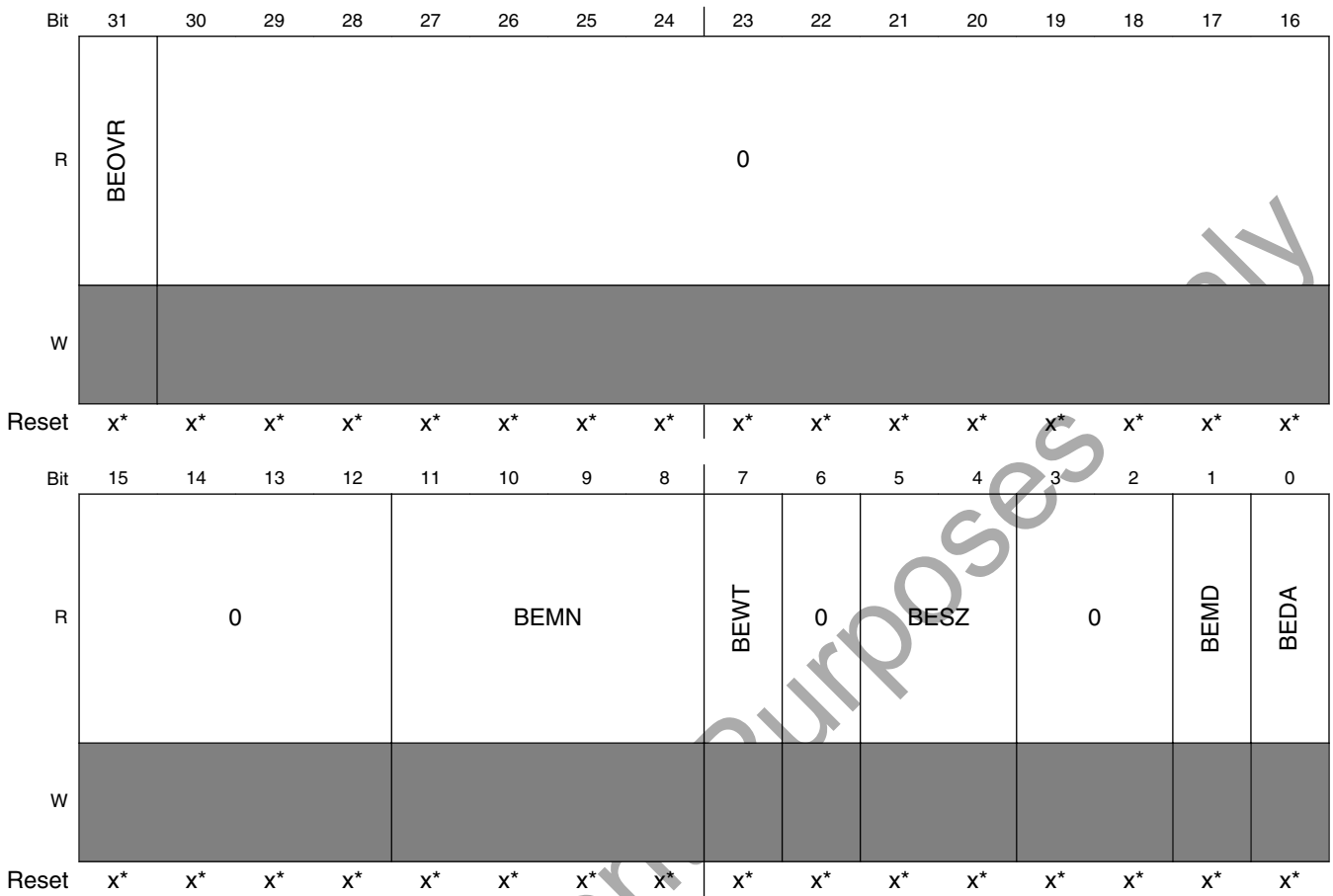
When a properly-enabled cache write buffer error interrupt event is detected, the faulting attributes are captured in the MCM_FATR register.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. Attempted writes have no effect.

For Assessment + Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 24h offset = 24h



* Notes:
- x = Undefined at reset.

MCM_FATR field descriptions

Field	Description
31 BEOVR	Bus error overrun Indicates if another cache write buffer bus error is detected before system software has retrieved all the error information from the original event. The window of time is defined from the detection of the original cache write buffer error termination until the MCM_ISCR[CWBER] is written with a 1 to clear it and rearm the capture logic. This bit is set by the hardware and cleared whenever software writes a 1 to the CWBER bit. 0 No bus error overrun 1 Bus error overrun occurred. The FADR and FDR registers and the other FATR bits are not updated to reflect this new bus error.
30–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 BEMN	Bus error master number Crossbar switch bus master number of the captured cache write buffer bus error. For this device, this value is always 0x1.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MCM_FATR field descriptions (continued)

Field	Description
7 BEWT	Bus error write Indicates the type of system bus access when the error was detected. Since this logic is monitoring data transfers from the cache write buffer, this bit is always a logical one, signaling a write operation. 0 Read access 1 Write access
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 BESZ	Bus error size Indicates the size of the cache write buffer access when the error was detected. 00 8-bit access 01 16-bit access 10 32-bit access 11 Reserved
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 BEMD	Bus error privilege level Indicates the privilege level of the cache write buffer access when the error was detected. 0 User mode 1 Supervisor/privileged mode
0 BEDA	Bus Error Data Access type Indicates the type of cache write buffer access when the error was detected. This attribute is always a logical one signaling a data reference. 0 Instruction 1 Data

For Assess

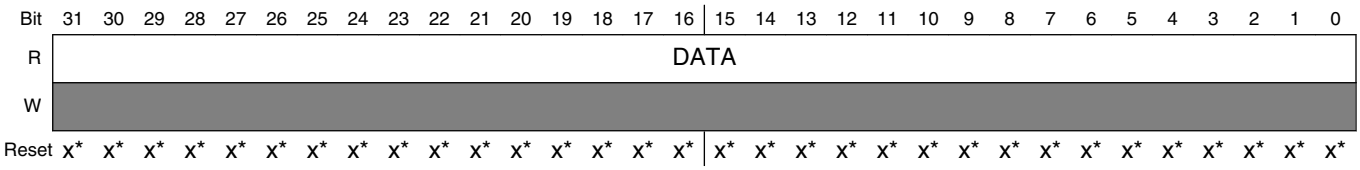
17.2.7 Store Buffer Fault Data Register (MCM_FDR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting data is captured in the MCM_FDR register.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. For byte and halfword writes, only the accessed byte lanes contain valid data; the contents of the other bytes are undefined. Attempted writes have no effect.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 28h offset = 28h



* Notes:
- x = Undefined at reset.

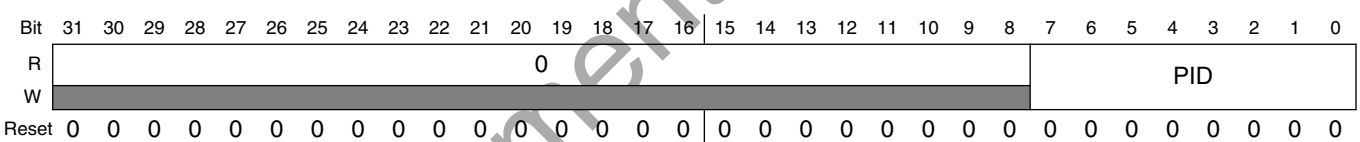
MCM_FDR field descriptions

Field	Description
DATA	Fault data

17.2.8 Process ID register (MCM_PID)

This register drives the M0_PID and M1_PID values in the Memory Protection Unit(MPU). System software loads this register before passing control to a given user mode process. If the PID of the process does not match the value in this register, a bus error occurs. See the MPU chapter for more details.

Address: 0h base + 30h offset = 30h



MCM_PID field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PID	M0_PID And M1_PID For MPU Drives the M0_PID and M1_PID values in the MPU.

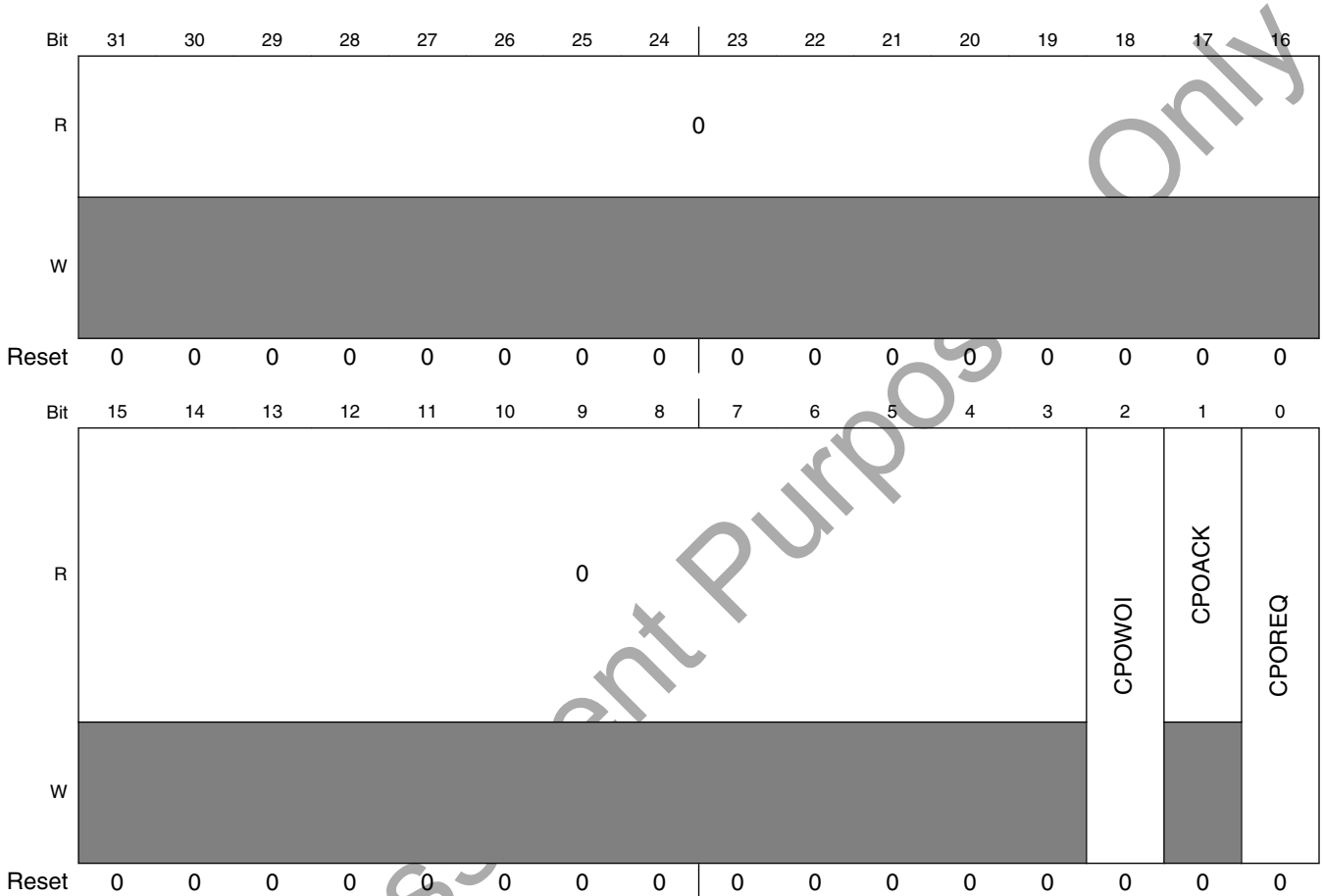
For Development Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

17.2.9 Compute Operation Control Register (MCM_CPO)

This register controls the Compute Operation.

Address: 0h base + 40h offset = 40h



Only
ent Purpos

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 wakeup 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...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MCM_CPO field descriptions (continued)

Field	Description
	0 Request is cleared. 1 Request Compute Operation.

17.2.10 Local Memory Descriptor Register (MCM_LMDR*n*)

NOTE

The LMDRn registers mapping to the LMEMs is as follows:

- LMDR0: SRAM_L
- LMDR1: SRAM_U
- LMDR2: PC CACHE

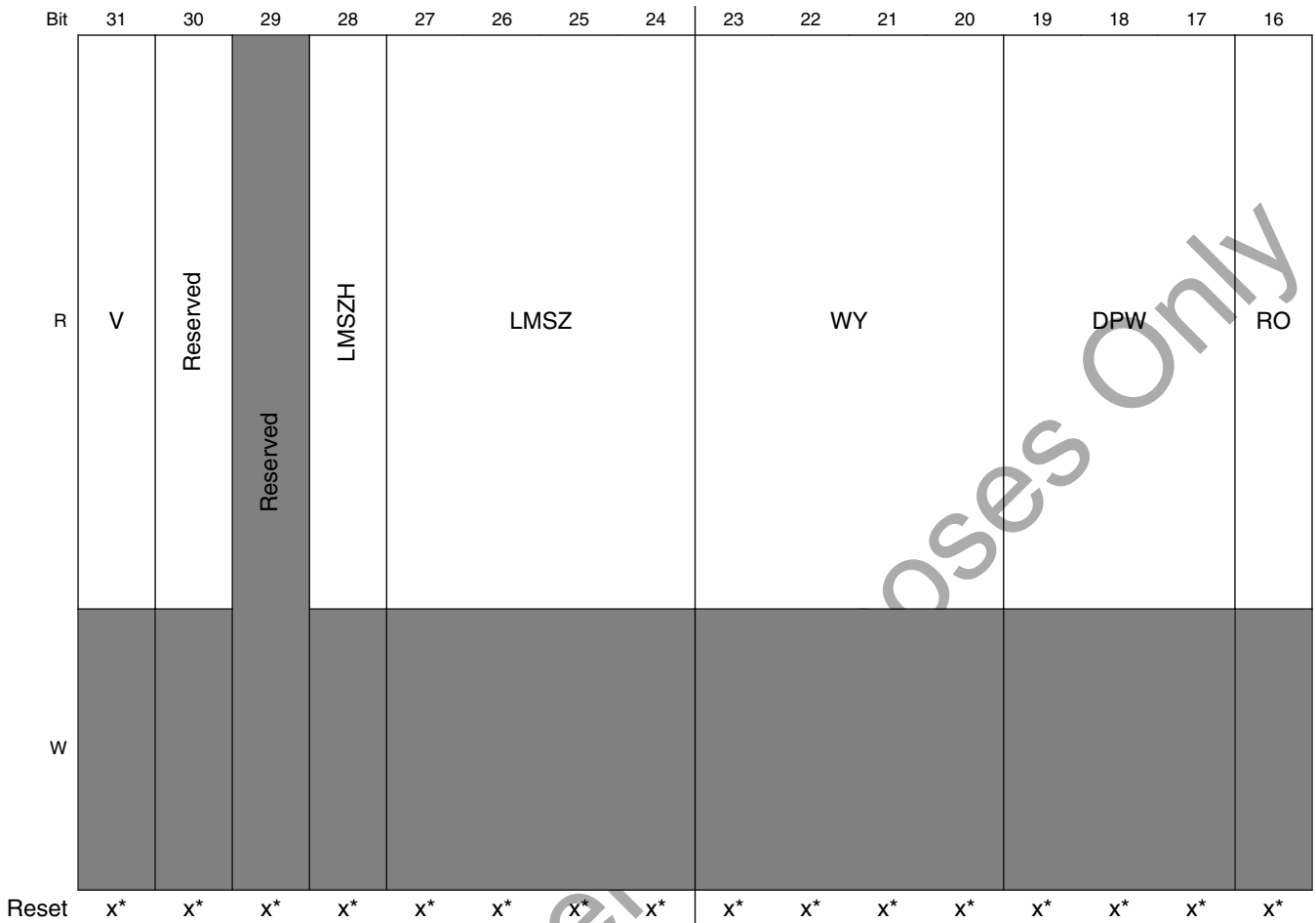
This section of the programming model is an array of 32-bit generic on-chip memory descriptor registers that provide static information on the attached memories as well as configurable controls (where appropriate).

Privileged 32-bit reads from a processor core or the debugger return the appropriate processor information. Reads from any other bus master return all zeroes. Privileged writes from a processor core or the debugger to writeable registers update the appropriate fields. Privileged writes from other bus masters are ignored. Attempted user mode accesses or any access with a size other than 32 bits are terminated with an error.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 400h offset + (4d × i), where i=0d to 2d



* Notes:

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- The reset values are different for the individual LMDR registers. LMDR0: 0x8604_0003; LMDR1: 0x8604_2003; LMDR2: 0x8424_40A0. x = Undefined at reset.

MCM_LMDR*n* field descriptions

Field	Description
31 V	Local memory Valid bit. This read-only field defines the validity (presence) of the local memory. 0 LMEMn is not present. 1 LMEMn is present.
30 Reserved	This field is reserved.
29 Reserved	This field is reserved.
28 LMSZH	LMEM Size “Hole”. For local memories that are not fully populated, that is, include a memory “hole” in the upper 25% of the address range, this bit is used. 0 LMEMn is a power-of-2 capacity. 1 LMEMn is not a power-of-2, with a capacity is 0.75 × LMSZ.
27–24 LMSZ	LMEM Size. This read-only field provides an encoded value of the local memory size. The capacity of the memory is expressed as Size [bytes] = $2^{(9+SZ)}$ where SZ is non-zero; a SZ = 0 indicates the memory is not present. 0000 no LMEMn (0 KB) 0001 1 KB LMEMn 0010 2 KB LMEMn 0011 4 KB LMEMn 0100 8 KB LMEMn 0101 16 KB LMEMn 0110 32 KB LMEMn 0111 64 KB LMEMn 1000 128 KB LMEMn 1001 256 KB LMEMn 1010 512 KB LMEMn 1011 1024 KB LMEMn 1100 2048 KB LMEMn 1101 4096 KB LMEMn 1110 8192 KB LMEMn 1111 16384 KB LMEMn
23–20 WY	Level 1 Cache Ways 0000 No Cache 0010 2-Way Set Associative 0100 4-Way Set Associative
19–17 DPW	LMEM Data Path Width. This read-only field defines the width of the local memory. 000-001 Reserved 010 LMEMn 32-bits wide 011 LMEMn 64-bits wide 100-111 Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MCM_LMDR*n* field descriptions (continued)

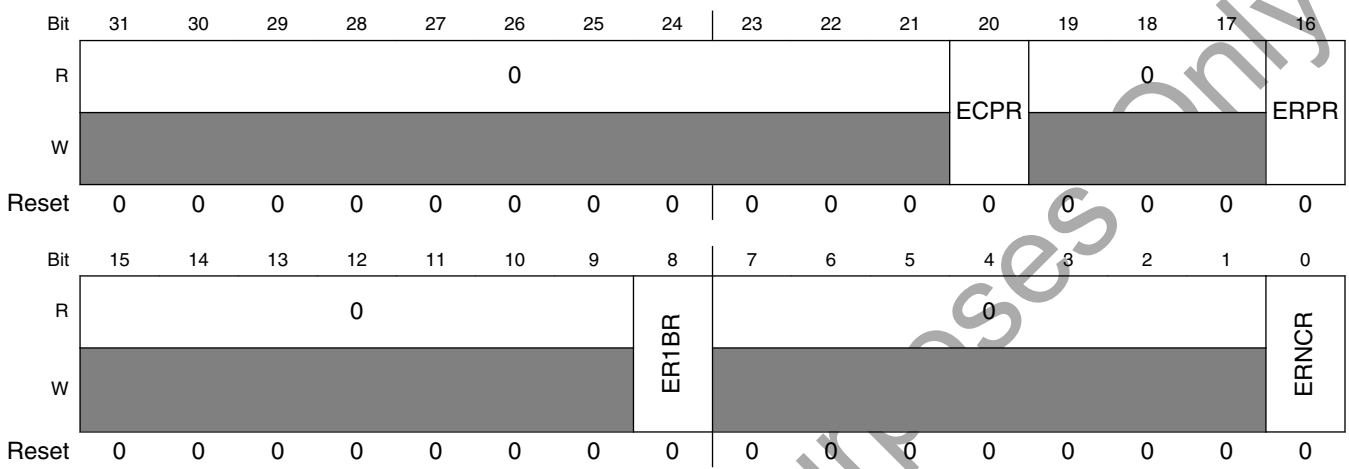
Field	Description
16 RO	Read-Only. This register bit provides a mechanism to “lock” the configuration state defined by LMDRn[7:0]. Once asserted, attempted writes to the LMDRn[7:0] register are ignored until the next reset clears the flag. 0 Writes to the LMDRn[7:0] are allowed. 1 Writes to the LMDRn[7:0] are ignored.
15–13 MT	Memory Type This field defines the type of the local memory. 000 SRAM_L 001 SRAM_U 010 PC Cache 011 PS Cache
12 Reserved	This field is reserved.
11–8 Reserved	This field is reserved.
7–4 CF1	Control Field 1 - for Cache Parity control functions • CF1[3] - PCPFE = PC Parity Fault Enable • CF1[2] - PSPFE = PS Parity Fault Enable • CF1[1] - PCPME = PC Parity Miss Enable • CF1[0] - PSPME = PS Parity Miss Enable
CF0	Control Field 0 - for TCM ECC control functions • CF0[3] - PFE = Parity Fault Enable • CF0[2] - RESERVED • CF0[1] - EERC = ECC Enable Read Check • CF0[0] - EEWG = ECC Enable Write Generation

For Assess

Freescale Confidential Proprietary - Non-Disclosure Agreement required

17.2.11 LMEM Parity & ECC Control Register (MCM_LMPECR)

Address: 0h base + 480h offset = 480h



MCM_LMPECR field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 ECPR	Enable Cache Parity Reporting 0 reporting enabled 1 reporting disabled
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 ERPR	Enable RAM Parity Reporting 0 reporting enabled 1 reporting disabled
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 ER1BR	Enable RAM ECC 1 Bit Reporting 0 reporting enabled 1 reporting disabled
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 ERNCR	Enable RAM ECC Noncorrectable Reporting 0 reporting enabled 1 reporting disabled

Freescale Confidential Proprietary Internal Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

17.2.12 LMEM Parity & ECC Interrupt Register (MCM_LMPEIR)

NOTE

Write 1 to the error bit in MCM_LMPEIR[23:0] can clear the interrupt flag.

Address: 0h base + 488h offset = 488h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V	0		PEELOC					PE							
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	E1B								ENC							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Purposes Only

MCM_LMPEIR field descriptions

Field	Description
31 V	Valid bit
30–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 PEELOC	Parity or ECC Error Location • 5'h00 - a non-correctable ECC event from SRAM_L • 5'h01 - a non-correctable ECC event from SRAM_U • 5'h08 - a 1-bit correctable ECC event from SRAM_L • 5'h09 - a 1-bit correctable ECC event from SRAM_U • 5'h14 - a PC Tag Parity Error • 5'h15 - a PC Data Parity Error
23–16 PE	Parity Error • [21] - PC Data Parity Error • [20] - PC Tag Parity Error • [19] - RESERVED • [18] - RESERVED
15–8 E1B	E1Bn = ECC 1-bit Error n • PEIR[15:10] - Reserved • PEIR[9] - 1-bit Error detected on SRAM_U • PEIR[8] - 1-bit Error detected on SRAM_L
ENC	ENCn = ECC Noncorrectable Error n

Table continues on the next page...

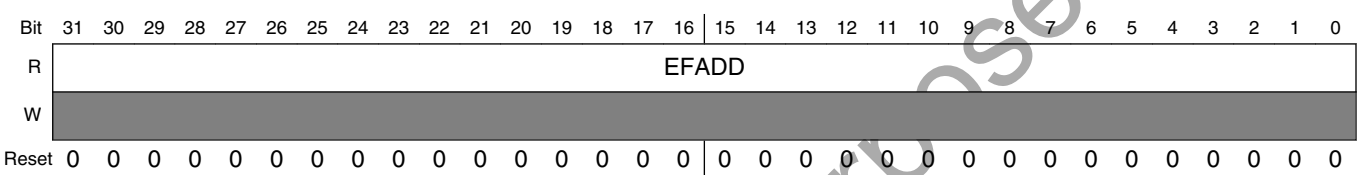
Freescale Confidential Proprietary - Non-Disclosure Agreement required

MCM_LMPEIR field descriptions (continued)

Field	Description
	• PEIR[7:2] - Reserved • PEIR[1] - Noncorrectable Error detected on SRAM_U • PEIR[0] - Noncorrectable Error detected on SRAM_L

17.2.13 LMEM Fault Address Register (MCM_LMFAR)

Address: 0h base + 490h offset = 490h



MCM_LMFAR field descriptions

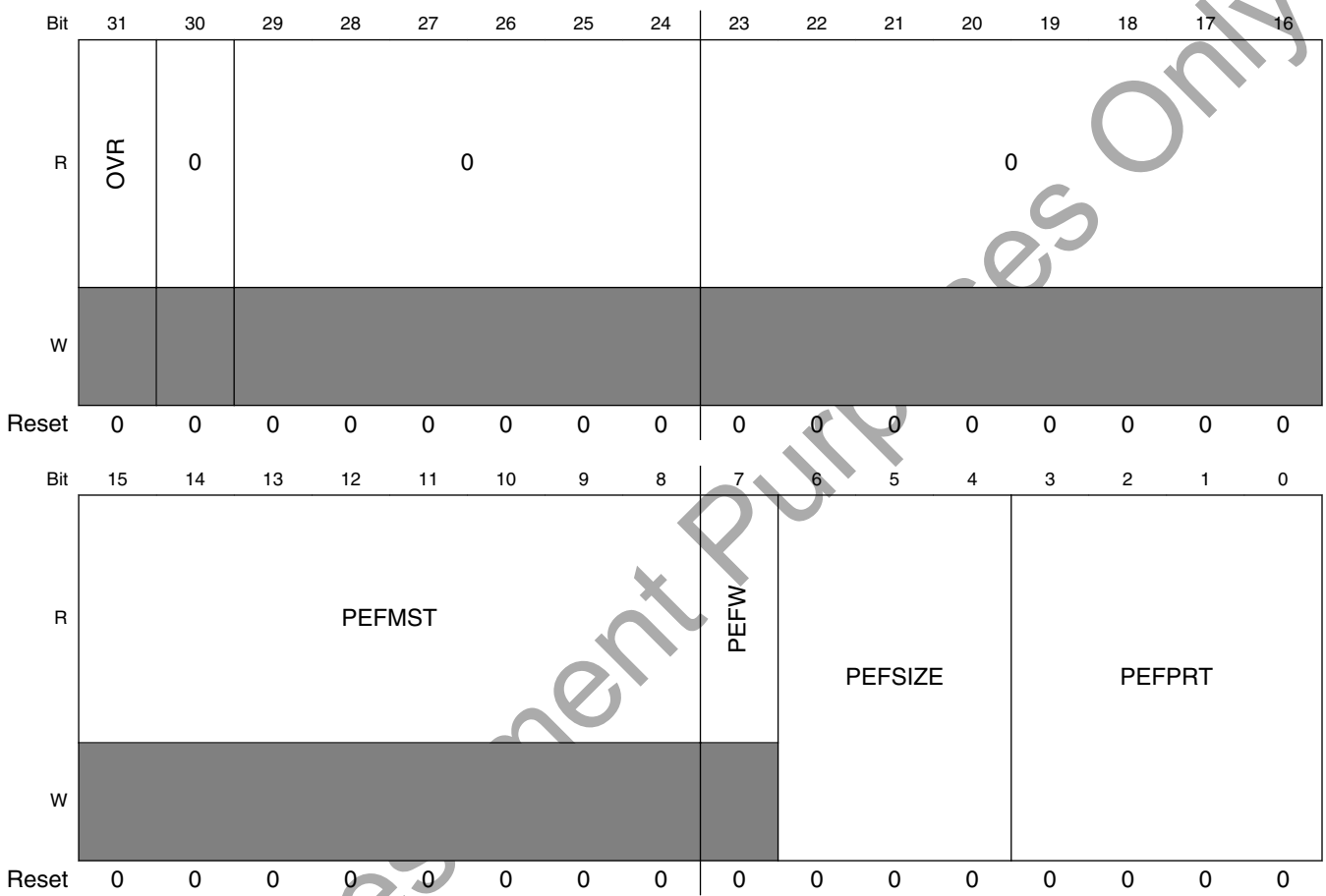
Field	Description
EFADD	ECC Fault Address

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

17.2.14 LMEM Fault Attribute Register (MCM_LMFATR)

Address: 0h base + 494h offset = 494h



Freescale Confidential Proprietary - Non-Disclosure Agreement Required for Review Purposes Only

MCM_LMFATR field descriptions

Field	Description
31 OVR	Overrun
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 PEFMST	Parity/ECC Fault Master Number
7 PEFW	Parity/ECC Fault Write

Table continues on the next page...

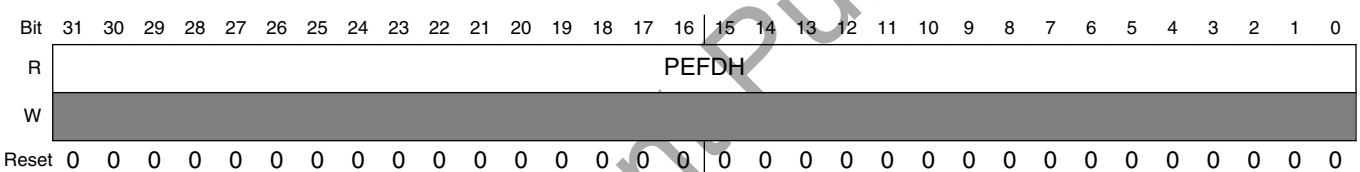
Freescale Confidential Proprietary - Non-Disclosure Agreement required

MCM_LMFATR field descriptions (continued)

Field	Description
6–4 PEFSIZE	Parity/ECC Fault Master Size • 3'b000 = 8-bit access • 3'b001 = 16-bit access • 3'b010 = 32-bit access • 3'b011 = 64-bit access • 3'b1xx = Reserved
PEFPRT	Parity/ECC Fault Protection • FATR[3] is Cacheable: 0=Non-cacheable, 1=Cacheable • FATR[2] is Bufferable: 0=Non-bufferable, 1=Bufferable • FATR[1] is Mode: 0=User mode, 1=Supervisor mode • FATR[0] is Type: 0=I-Fetch, 1=Data

17.2.15 LMEM Fault Data High Register (MCM_LMFDHR)

Address: 0h base + 4A0h offset = 4A0h

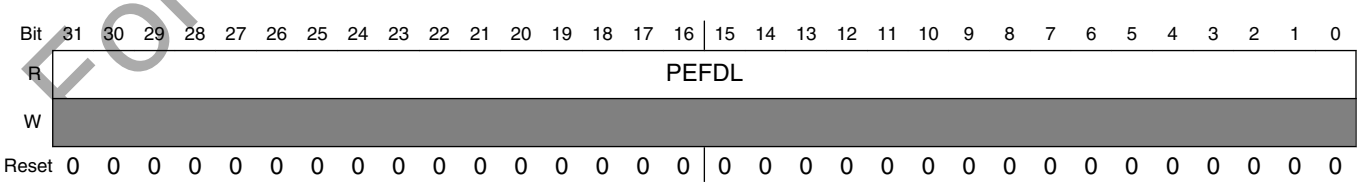


MCM_LMFDHR field descriptions

Field	Description
PEFDH	Parity or ECC Fault Data High

17.2.16 LMEM Fault Data Low Register (MCM_LMFDLR)

Address: 0h base + 4A4h offset = 4A4h



MCM_LMFDLR field descriptions

Field	Description
PEFDL	Parity or ECC Fault Data Low

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MCM_LMFDLR field descriptions (continued)

Field	Description

17.3 Functional description

This section describes the functional description of MCM module.

17.3.1 Interrupts

The MCM's interrupt is generated if any of the following is true:

- FPU input denormal interrupt is enabled (FIDCE) and an input is denormalized (FIDC)
- FPU inexact interrupt is enabled (FIXCE) and a number is inexact (FIXC)
- FPU underflow interrupt is enabled (FUFCE) and an underflow occurs (FUFC)
- FPU overflow interrupt is enabled (FOFCE) and an overflow occurs (FOFC)
- FPU divide-by-zero interrupt is enabled (FDZCE) and a divide-by-zero occurs (FDZC)
- FPU invalid operation interrupt is enabled (FIOCE) and an invalid occurs (FIOC)

- SRAM_L correctable(1-bit) ECC error
- SRAM_L uncorrectable ECC error
- SRAM_U correctable (1-bit) ECC error
- SRAM_U uncorrectable ECC error
- PC data parity error
- PC tag parity error
- Cache write buffer error

17.3.1.1 Determining source of the interrupt

To determine the exact source of the interrupt qualify the interrupt status flags with the corresponding interrupt enable bits.

1. From MCM_ISCR[31:16] && MCM_ISCR[15:0]
2. Search the result for asserted flags, which indicate the exact interrupt sources

NOTE

ECC and Parity interrupts are determined by LMEPECR (interrupt enable) and LMPEIR (interrupt source).

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 18
Crossbar Switch Lite (AXBS-Lite)

18.1 Chip-specific AXBS information

18.1.1 Crossbar Switch Master Assignments

The masters connected to the crossbar switch are assigned as in the below given table.

Master module	Master port number	Priority in fixed priority mode [1]
ARM core code bus	0	Lowest
ARM core system bus	1	
DMA	2	Highest

1. The selection of the global slave port arbitration is controlled by MCM module. For fixed priority, set MCM _CPCR[CBRR] to 0. For round robin, set MCM_CPCR[CBRR]. The arbitration setting applies to all slave ports.

18.1.2 Crossbar Switch Slave Assignments

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

Slave module	Slave port number	Protected by MPU
Flash memory controller	0	Yes
SRAM controllers	1	Yes
Peripheral bridge 0 / GPIO	2	No. Protection built into Peripheral Bridge (AIPS-Lite).

18.2 Introduction

The information found here provides information on the layout, configuration, and programming of the crossbar switch.

For Assessment Purposes Only

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.

18.2.1 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).

18.3 Memory Map / Register Definition

This crossbar switch is designed for minimal gate count. It, therefore, has no memory-mapped configuration registers.

Please see the chip-specific information for information on whether the arbitration method in the crossbar switch is programmable, and by which module.

18.4 Functional Description

18.4.1 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

18.4.2 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.

18.4.2.1 Arbitration during undefined length bursts

All lengths of burst accesses lock out arbitration until the last beat of the burst.

18.4.2.2 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.

For Assessment Purposes Only

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 18-1. How the Crossbar Switch grants control of a slave port to a master

When	Then the Crossbar Switch grants control to the requesting master
Both of the following are true: • The current master is not running a transfer. • The new requesting master's priority level is higher than that of the current master.	At the next clock edge
The requesting master's priority level is lower than the current master.	At the conclusion of one of the following cycles: • An IDLE cycle • A non-IDLE cycle to a location other than the current slave port

18.4.2.3 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

18.5 Initialization/application information

No initialization is required for the crossbar switch.

See the AXBS section of the configuration chapter for the reset state of the arbitration scheme.

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 19
Memory Protection Unit (MPU)

19.1 Chip-specific MPU information

19.1.1 MPU Slave Port Assignments

The memory-mapped resources protected by the MPU are:

Table 19-1. MPU Slave Port Assignments

Source	MPU Slave Port Assignment	Destination
Crossbar slave port 0	MPU slave port 0	Flash Controller and boot ROM
Crossbar slave port 1	MPU slave port 1	SRAM backdoor
Code Bus	MPU slave port 2	SRAM_L frontdoor
System Bus	MPU slave port 3	SRAM_U frontdoor

19.1.2 MPU Logical Bus Master Assignments

The logical bus master assignments for the MPU are:

Table 19-2. MPU Logical Bus Master Assignments

MPU Logical Bus Master ID	Bus Master
0	Core
1	Debugger
2	DMA

Freescale Confidential Proprietary Non-Disclosure Agreement required

Freescale Confidential Proprietary - Non-Disclosure Agreement required

19.1.3 MPU register reset values

The following table lists MPU register reset values that are specific to this chip. For the reset values of all other MPU registers, see Memory map/register definition.

Table 19-3. Chip-specific MPU register reset values

Register	Reset value	Notes
CESR	0081_4200h	NSP field resets to 0100b

19.2 Introduction

The memory protection unit (MPU) provides hardware access control for all memory references generated in the device.

19.3 Overview

The MPU concurrently monitors all system bus transactions and evaluates their appropriateness using pre-programmed region descriptors that define memory spaces and their access rights. Memory references that have sufficient access control rights are allowed to complete, while references that are not mapped to any region descriptor or have insufficient rights are terminated with a protection error response.

19.3.1 Block diagram

A simplified block diagram of the MPU module is shown in the following figure.

For Assessment Purposes C

Freescale Confidential Proprietary - Non-Disclosure Agreement required

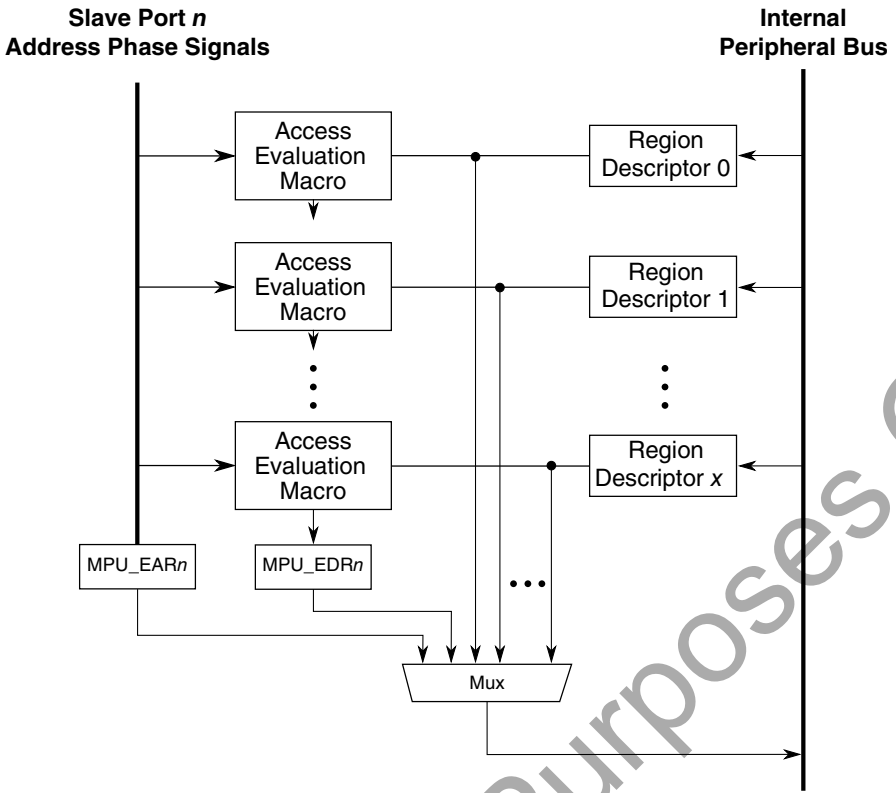


Figure 19-1. MPU block diagram

The hardware's two-dimensional connection matrix is clearly visible with the basic access evaluation macro shown as the replicated submodule block. The crossbar switch slave ports are shown on the left, the region descriptor registers in the middle, and the peripheral bus interface on the right side. The evaluation macro contains two magnitude comparators connected to the start and end address registers from each region descriptor as well as the combinational logic blocks to determine the region hit and the access protection error. For details of the access evaluation macro, see Access evaluation macro.

19.3.2 Features

The MPU implements a two-dimensional hardware array of memory region descriptors and the crossbar slave ports to continuously monitor the legality of every memory reference generated by each bus master in the system.

The feature set includes:

- 8 program-visible 128-bit region descriptors, accessible by four 32-bit words each
  - Each region descriptor defines a modulo-32 byte space, aligned anywhere in memory

For Assessment Purposes Only

    - Region sizes can vary from 32 bytes to 4 Gbytes
  - Two access control permissions defined in a single descriptor word
    - Masters 0–3: read, write, and execute attributes for supervisor and user accesses
    - Masters 4–7: read and write attributes
  - Hardware-assisted maintenance of the descriptor valid bit minimizes coherency issues
  - Alternate programming model view of the access control permissions word
  - Priority given to granting permission over denying access for overlapping region descriptors
- Detects access protection errors if a memory reference does not hit in any memory region, or if the reference is illegal in all hit memory regions. If an access error occurs, the reference is terminated with an error response, and the MPU inhibits the bus cycle being sent to the targeted slave device.
- Error registers, per slave port, capture the last faulting address, attributes, and other information
- Global MPU enable/disable control bit

19.4 Memory map/register definition

The programming model is partitioned into three groups:

- Control/status registers
- The data structure containing the region descriptors
- The alternate view of the region descriptor access control values

The programming model can only be referenced using 32-bit accesses. Attempted references using different access sizes, to undefined, that is, reserved, addresses, or with a non-supported access type, such as a write to a read-only register, or a read of a write-only register, generate an error termination.

The programming model can be accessed only in supervisor mode.

NOTE

See the chip configuration details for any chip-specific register information in this module.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MPU memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Control/Error Status Register (MPU_CESR)	32	R/W	0081_4001h	19.4.1/258
10	Error Address Register, slave port n (MPU_EAR0)	32	R	0000_0000h	19.4.2/260
14	Error Detail Register, slave port n (MPU_EDR0)	32	R	0000_0000h	19.4.3/260
18	Error Address Register, slave port n (MPU_EAR1)	32	R	0000_0000h	19.4.2/260
1C	Error Detail Register, slave port n (MPU_EDR1)	32	R	0000_0000h	19.4.3/260
20	Error Address Register, slave port n (MPU_EAR2)	32	R	0000_0000h	19.4.2/260
24	Error Detail Register, slave port n (MPU_EDR2)	32	R	0000_0000h	19.4.3/260
28	Error Address Register, slave port n (MPU_EAR3)	32	R	0000_0000h	19.4.2/260
2C	Error Detail Register, slave port n (MPU_EDR3)	32	R	0000_0000h	19.4.3/260
400	Region Descriptor n, Word 0 (MPU_RGD0_WORD0)	32	R/W	0000_0000h	19.4.4/261
404	Region Descriptor n, Word 1 (MPU_RGD0_WORD1)	32	R/W	See section	19.4.5/262
408	Region Descriptor n, Word 2 (MPU_RGD0_WORD2)	32	R/W	See section	19.4.6/262
40C	Region Descriptor n, Word 3 (MPU_RGD0_WORD3)	32	R/W	See section	19.4.7/265
410	Region Descriptor n, Word 0 (MPU_RGD1_WORD0)	32	R/W	0000_0000h	19.4.4/261
414	Region Descriptor n, Word 1 (MPU_RGD1_WORD1)	32	R/W	See section	19.4.5/262
418	Region Descriptor n, Word 2 (MPU_RGD1_WORD2)	32	R/W	See section	19.4.6/262
41C	Region Descriptor n, Word 3 (MPU_RGD1_WORD3)	32	R/W	See section	19.4.7/265
420	Region Descriptor n, Word 0 (MPU_RGD2_WORD0)	32	R/W	0000_0000h	19.4.4/261
424	Region Descriptor n, Word 1 (MPU_RGD2_WORD1)	32	R/W	See section	19.4.5/262
428	Region Descriptor n, Word 2 (MPU_RGD2_WORD2)	32	R/W	See section	19.4.6/262
42C	Region Descriptor n, Word 3 (MPU_RGD2_WORD3)	32	R/W	See section	19.4.7/265
430	Region Descriptor n, Word 0 (MPU_RGD3_WORD0)	32	R/W	0000_0000h	19.4.4/261
434	Region Descriptor n, Word 1 (MPU_RGD3_WORD1)	32	R/W	See section	19.4.5/262
438	Region Descriptor n, Word 2 (MPU_RGD3_WORD2)	32	R/W	See section	19.4.6/262
43C	Region Descriptor n, Word 3 (MPU_RGD3_WORD3)	32	R/W	See section	19.4.7/265
440	Region Descriptor n, Word 0 (MPU_RGD4_WORD0)	32	R/W	0000_0000h	19.4.4/261
444	Region Descriptor n, Word 1 (MPU_RGD4_WORD1)	32	R/W	See section	19.4.5/262
448	Region Descriptor n, Word 2 (MPU_RGD4_WORD2)	32	R/W	See section	19.4.6/262
44C	Region Descriptor n, Word 3 (MPU_RGD4_WORD3)	32	R/W	See section	19.4.7/265
450	Region Descriptor n, Word 0 (MPU_RGD5_WORD0)	32	R/W	0000_0000h	19.4.4/261
454	Region Descriptor n, Word 1 (MPU_RGD5_WORD1)	32	R/W	See section	19.4.5/262
458	Region Descriptor n, Word 2 (MPU_RGD5_WORD2)	32	R/W	See section	19.4.6/262
45C	Region Descriptor n, Word 3 (MPU_RGD5_WORD3)	32	R/W	See section	19.4.7/265
460	Region Descriptor n, Word 0 (MPU_RGD6_WORD0)	32	R/W	0000_0000h	19.4.4/261
464	Region Descriptor n, Word 1 (MPU_RGD6_WORD1)	32	R/W	See section	19.4.5/262
468	Region Descriptor n, Word 2 (MPU_RGD6_WORD2)	32	R/W	See section	19.4.6/262
46C	Region Descriptor n, Word 3 (MPU_RGD6_WORD3)	32	R/W	See section	19.4.7/265
470	Region Descriptor n, Word 0 (MPU_RGD7_WORD0)	32	R/W	0000_0000h	19.4.4/261
474	Region Descriptor n, Word 1 (MPU_RGD7_WORD1)	32	R/W	See section	19.4.5/262

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MPU memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
478	Region Descriptor n, Word 2 (MPU_RGD7_WORD2)	32	R/W	See section	19.4.6/262
47C	Region Descriptor n, Word 3 (MPU_RGD7_WORD3)	32	R/W	See section	19.4.7/265
800	Region Descriptor Alternate Access Control n (MPU_RGDAAC0)	32	R/W	See section	19.4.8/266
804	Region Descriptor Alternate Access Control n (MPU_RGDAAC1)	32	R/W	See section	19.4.8/266
808	Region Descriptor Alternate Access Control n (MPU_RGDAAC2)	32	R/W	See section	19.4.8/266
80C	Region Descriptor Alternate Access Control n (MPU_RGDAAC3)	32	R/W	See section	19.4.8/266
810	Region Descriptor Alternate Access Control n (MPU_RGDAAC4)	32	R/W	See section	19.4.8/266
814	Region Descriptor Alternate Access Control n (MPU_RGDAAC5)	32	R/W	See section	19.4.8/266
818	Region Descriptor Alternate Access Control n (MPU_RGDAAC6)	32	R/W	See section	19.4.8/266
81C	Region Descriptor Alternate Access Control n (MPU_RGDAAC7)	32	R/W	See section	19.4.8/266

19.4.1 Control/Error Status Register (MPU_CESR)

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SPERR				0				1	0			HRL			
W	w1c															
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

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

MPU_CESR field descriptions

Field	Description
31–28 SPERR	Slave Port n Error Indicates a captured error in EARn and EDRn. This bit is set when the hardware detects an error and records the faulting address and attributes. It is cleared by writing one to it. If another error is captured at the exact same cycle as the write, the flag remains set. A find-first-one instruction or equivalent can detect the presence of a captured error. The following shows the correspondence between the bit number and slave port number: • Bit 31 corresponds to slave port 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MPU_CESR field descriptions (continued)

Field	Description
	• Bit 30 corresponds to slave port 1. • Bit 29 corresponds to slave port 2. • Bit 28 corresponds to slave port 3. 0 No error has occurred for slave port n. 1 An error has occurred for slave port n.
27–24 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 1.
22–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 HRL	Hardware Revision Level Specifies the MPU's hardware and definition revision level. It can be read by software to determine the functional definition of the module.
15–12 NSP	Number Of Slave Ports Specifies the number of slave ports connected to the MPU.
11–8 NRGD	Number Of Region Descriptors Indicates the number of region descriptors implemented in the MPU. 0000 8 region descriptors 0001 12 region descriptors 0010 16 region descriptors
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 VLD	Valid Global enable/disable for the MPU. 0 MPU is disabled. All accesses from all bus masters are allowed. 1 MPU is enabled

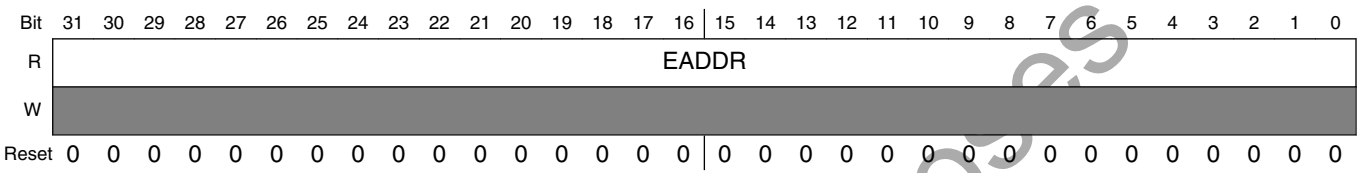
For Asse

Freescale Confidential Proprietary - Non-Disclosure Agreement required

19.4.2 Error Address Register, slave port n (MPU_EAR*n*)

When the MPU detects an access error on slave port n, the 32-bit reference address is captured in this read-only register and the corresponding bit in CESR[SPERR] set. Additional information about the faulting access is captured in the corresponding EDRn at the same time. This register and the corresponding EDRn contain the most recent access error; there are no hardware interlocks with CESR[SPERR], as the error registers are always loaded upon the occurrence of each protection violation.

Address: 0h base + 10h offset + (8d × i), where i=0d to 3d



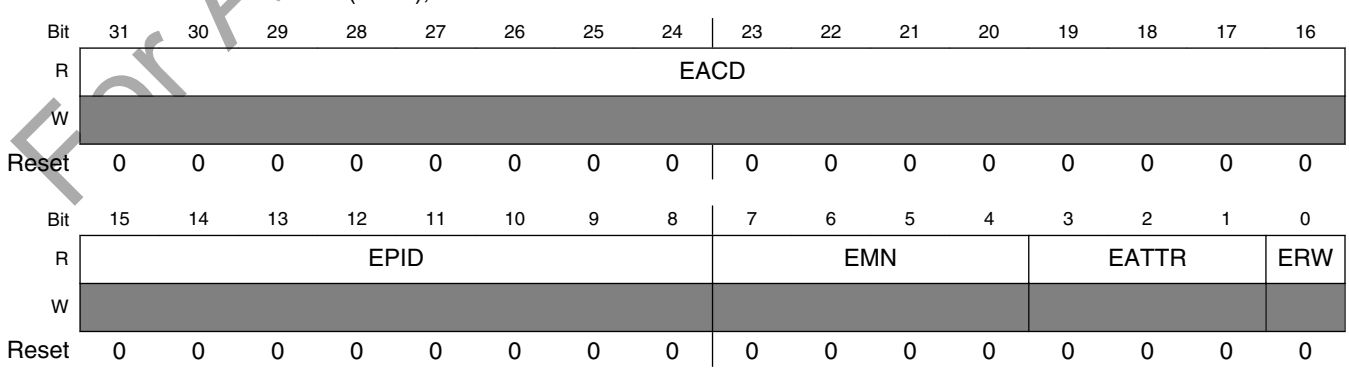
MPU_EAR*n* field descriptions

Field	Description
EADDR	Error Address Indicates the reference address from slave port n that generated the access error

19.4.3 Error Detail Register, slave port n (MPU_EDR*n*)

When the MPU detects an access error on slave port n, 32 bits of error detail are captured in this read-only register and the corresponding bit in CESR[SPERR] is set. Information on the faulting address is captured in the corresponding EARn register at the same time. This register and the corresponding EARn register contain the most recent access error; there are no hardware interlocks with CESR[SPERR] as the error registers are always loaded upon the occurrence of each protection violation.

Address: 0h base + 14h offset + (8d × i), where i=0d to 3d



For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MPU_EDR*n* field descriptions

Field	Description
31–16 EACD	Error Access Control Detail Indicates the region descriptor with the access error. • If EDRn contains a captured error and EACD is cleared, an access did not hit in any region descriptor. • If only a single EACD bit is set, the protection error was caused by a single non-overlapping region descriptor. • If two or more EACD bits are set, the protection error was caused by an overlapping set of region descriptors.
15–8 EPID	Error Process Identification Records the process identifier of the faulting reference. The process identifier is typically driven only by processor cores; for other bus masters, this field is cleared.
7–4 EMN	Error Master Number Indicates the bus master that generated the access error.
3–1 EATTR	Error Attributes Indicates attribute information about the faulting reference. **NOTE:** All other encodings are reserved. 000 User mode, instruction access 001 User mode, data access 010 Supervisor mode, instruction access 011 Supervisor mode, data access
0 ERW	Error Read/Write Indicates the access type of the faulting reference. 0 Read 1 Write

For Assess

19.4.4 Region Descriptor n, Word 0 (MPU_RGD*n*_WORD0)

The first word of the region descriptor defines the 0-modulo-32 byte start address of the memory region. Writes to this register clear the region descriptor's valid bit (RGDn_WORD3[VLD]).

Address: 0h base + 400h offset + (16d × 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	SRTADDR																											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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MPU_RGD*n*_WORD0 field descriptions

Field	Description
31–5 SRTADDR	Start Address Defines the most significant bits of the 0-modulo-32 byte start address of the memory region.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

19.4.5 Region Descriptor n, Word 1 (MPU_RGD*n*_WORD1)

The second word of the region descriptor defines the 31-modulo-32 byte end address of the memory region. Writes to this register clear the region descriptor's valid bit (RGDn_WORD3[VLD]).

Address: 0h base + 404h offset + (16d × 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	ENDADDR																											Reserved				
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	1*	1*	1*	1*	1*

* Notes:
- Reset value of RGD0_WORD1 is FFFF_FFFFh

  Reset value of RGD[1:7]_WORD1 is 0000_001Fh

MPU_RGD*n*_WORD1 field descriptions

Field	Description
31–5 ENDADDR	End Address Defines the most significant bits of the 31-modulo-32 byte end address of the memory region. **NOTE:** The MPU does not verify that ENDADDR ≥ SRTADDR.
Reserved	This field is reserved.

19.4.6 Region Descriptor n, Word 2 (MPU_RGD*n*_WORD2)

The third word of the region descriptor defines the access control rights of the memory region. The access control privileges depend on two broad classifications of bus masters:

- Bus masters 0–3 have a 5-bit field defining separate privilege rights for user and supervisor mode accesses, as well as the optional inclusion of a process identification field within the definition.
- Bus masters 4–7 are limited to separate read and write permissions.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For the privilege rights of bus masters 0–3, there are three flags associated with this function:

- Read (r) refers to accessing the referenced memory address using an operand (data) fetch
- Write (w) refers to updating the referenced memory address using a store (data) instruction
- Execute (x) refers to reading the referenced memory address using an instruction fetch

Writes to RGDn_WORD2 clear the region descriptor's valid bit (RGDn_WORD3[VLD]). If only updating the access controls, write to RGDAACn instead because stores to these locations do not affect the descriptor's valid bit.

Address: 0h base + 408h offset + (16d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	M7RE	M7WE	M6RE	M6WE	M5RE	M5WE	M4RE	M4WE	M3PE	M3SM		M3UM			M2PE	M2SM
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	M2SM	M2UM			M1PE	M1SM			M1UM		M0PE	M0SM			M0UM	
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Reset value of RGD0_WORD2 is 0061_F7DFh

  Reset value of RGD[1:7]_WORD2 is 0000_0000h

Assessment Purposes Only

MPU_RGD*n*_WORD2 field descriptions

Field	Description
31 M7RE	Bus Master 7 Read Enable 0 Bus master 7 reads terminate with an access error and the read is not performed 1 Bus master 7 reads allowed
30 M7WE	Bus Master 7 Write Enable 0 Bus master 7 writes terminate with an access error and the write is not performed 1 Bus master 7 writes allowed
29 M6RE	Bus Master 6 Read Enable 0 Bus master 6 reads terminate with an access error and the read is not performed 1 Bus master 6 reads allowed

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MPU_RGD*n*_WORD2 field descriptions (continued)

Field	Description
28 M6WE	Bus Master 6 Write Enable 0 Bus master 6 writes terminate with an access error and the write is not performed 1 Bus master 6 writes allowed
27 M5RE	Bus Master 5 Read Enable 0 Bus master 5 reads terminate with an access error and the read is not performed 1 Bus master 5 reads allowed
26 M5WE	Bus Master 5 Write Enable 0 Bus master 5 writes terminate with an access error and the write is not performed 1 Bus master 5 writes allowed
25 M4RE	Bus Master 4 Read Enable 0 Bus master 4 reads terminate with an access error and the read is not performed 1 Bus master 4 reads allowed
24 M4WE	Bus Master 4 Write Enable 0 Bus master 4 writes terminate with an access error and the write is not performed 1 Bus master 4 writes allowed
23 M3PE	Bus Master 3 Process Identifier Enable 0 Do not include the process identifier in the evaluation 1 Include the process identifier and mask (RGDn_WORD3) in the region hit evaluation
22–21 M3SM	Bus Master 3 Supervisor Mode Access Control Defines the access controls for bus master 3 in Supervisor mode. 00 r/w/x; read, write and execute allowed 01 r/x; read and execute allowed, but no write 10 r/w; read and write allowed, but no execute 11 Same as User mode defined in M3UM
20–18 M3UM	Bus Master 3 User Mode Access Control Defines the access controls for bus master 3 in User mode. M3UM consists of three independent bits, enabling read (r), write (w), and execute (x) permissions. In M3UM[2:0]: M3UM[2] controls read permissions, M3UM[1] controls write permissions, and M3UM[0] controls execute permissions. 0 An attempted access of that mode may be terminated with an access error (if not allowed by another descriptor) and the access not performed. 1 Allows the given access type to occur
17 M2PE	Bus Master 2 Process Identifier Enable See M3PE description.
16–15 M2SM	Bus Master 2 Supervisor Mode Access Control See M3SM description.
14–12 M2UM	Bus Master 2 User Mode Access control See M3UM description.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MPU_RGD*n*_WORD2 field descriptions (continued)

Field	Description
11 M1PE	Bus Master 1 Process Identifier enable See M3PE description.
10–9 M1SM	Bus Master 1 Supervisor Mode Access Control See M3SM description.
8–6 M1UM	Bus Master 1 User Mode Access Control See M3UM description.
5 M0PE	Bus Master 0 Process Identifier enable See M0PE description.
4–3 M0SM	Bus Master 0 Supervisor Mode Access Control See M3SM description.
M0UM	Bus Master 0 User Mode Access Control See M3UM description.

19.4.7 Region Descriptor n, Word 3 (MPU_RGD*n*_WORD3)

The fourth word of the region descriptor contains the optional process identifier and mask, plus the region descriptor's valid bit.

Address: 0h base + 40Ch offset + (16d × i), where i=0d to 7d

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

* Notes:
- Reset value of RGD0_WORD3 is 0000_0001h

  Reset value of RGD[1:7]_WORD3 is 0000_0000h

MPU_RGD*n*_WORD3 field descriptions

Field	Description
31–24 PID	Process Identifier

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purp

MPU_RGD*n*_WORD3 field descriptions (continued)

Field	Description
	Specifies the process identifier that is included in the region hit determination if RGDn_WORD2[MxPE] is set. PIDMASK can mask individual bits in this field.
23–16 PIDMASK	Process Identifier Mask Provides a masking capability so that multiple process identifiers can be included as part of the region hit determination. If a bit in PIDMASK is set, then the corresponding PID bit is ignored in the comparison. This field and PID are included in the region hit determination if RGDn_WORD2[MxPE] is set. For more information on the handling of the PID and PIDMASK, see “Access Evaluation - Hit Determination.”
15–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 VLD	Valid Signals the region descriptor is valid. Any write to RGDn_WORD0–2 clears this bit. 0 Region descriptor is invalid 1 Region descriptor is valid

19.4.8 Region Descriptor Alternate Access Control n (MPU_RGDAAC*n*)

Because software may adjust only the access controls within a region descriptor (RGDn_WORD2) as different tasks execute, an alternate programming view of this 32-bit entity is available. Writing to this register does not affect the descriptor’s valid bit.

Address: 0h base + 800h 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
R W	M7RE	M7WE	M6RE	M6WE	M5RE	M5WE	M4RE	M4WE	M3PE	M3SM		M3UM			M2PE	M2SM
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	M2SM	M2UM			M1PE	M1SM		M1UM			M0PE	M0SM		M0UM		
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Reset value of RGDAAC0 is 0061_F7DFh

  Reset value of RGDAAC[1:7] is 0000_0000h

For Assessment Purpo

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MPU_RGDAAC*n* field descriptions

Field	Description
31 M7RE	Bus Master 7 Read Enable 0 Bus master 7 reads terminate with an access error and the read is not performed 1 Bus master 7 reads allowed
30 M7WE	Bus Master 7 Write Enable 0 Bus master 7 writes terminate with an access error and the write is not performed 1 Bus master 7 writes allowed
29 M6RE	Bus Master 6 Read Enable 0 Bus master 6 reads terminate with an access error and the read is not performed 1 Bus master 6 reads allowed
28 M6WE	Bus Master 6 Write Enable 0 Bus master 6 writes terminate with an access error and the write is not performed 1 Bus master 6 writes allowed
27 M5RE	Bus Master 5 Read Enable 0 Bus master 5 reads terminate with an access error and the read is not performed 1 Bus master 5 reads allowed
26 M5WE	Bus Master 5 Write Enable 0 Bus master 5 writes terminate with an access error and the write is not performed 1 Bus master 5 writes allowed
25 M4RE	Bus Master 4 Read Enable 0 Bus master 4 reads terminate with an access error and the read is not performed 1 Bus master 4 reads allowed
24 M4WE	Bus Master 4 Write Enable 0 Bus master 4 writes terminate with an access error and the write is not performed 1 Bus master 4 writes allowed
23 M3PE	Bus Master 3 Process Identifier Enable 0 Do not include the process identifier in the evaluation 1 Include the process identifier and mask (RGDn.RGDAAC) in the region hit evaluation
22–21 M3SM	Bus Master 3 Supervisor Mode Access Control Defines the access controls for bus master 3 in Supervisor mode. 00 r/w/x; read, write and execute allowed 01 r/x; read and execute allowed, but no write 10 r/w; read and write allowed, but no execute 11 Same as User mode defined in M3UM
20–18 M3UM	Bus Master 3 User Mode Access Control Defines the access controls for bus master 3 in user mode. M3UM consists of three independent bits, enabling read (r), write (w), and execute (x) permissions. In M3UM[2:0]: M3UM[2] controls read permissions, M3UM[1] controls write permissions, and M3UM[0] controls execute permissions.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MPU_RGDAAC*n* field descriptions (continued)

Field	Description
	0 An attempted access of that mode may be terminated with an access error (if not allowed by another descriptor) and the access not performed. 1 Allows the given access type to occur
17 M2PE	Bus Master 2 Process Identifier Enable See M3PE description.
16–15 M2SM	Bus Master 2 Supervisor Mode Access Control See M3SM description.
14–12 M2UM	Bus Master 2 User Mode Access Control See M3UM description.
11 M1PE	Bus Master 1 Process Identifier Enable See M3PE description.
10–9 M1SM	Bus Master 1 Supervisor Mode Access Control See M3SM description.
8–6 M1UM	Bus Master 1 User Mode Access Control See M3UM description.
5 M0PE	Bus Master 0 Process Identifier Enable See M3PE description.
4–3 M0SM	Bus Master 0 Supervisor Mode Access Control See M3SM description.
M0UM	Bus Master 0 User Mode Access Control See M3UM description.

For Assess

19.5 Functional description

In this section, the functional operation of the MPU is detailed, including the operation of the access evaluation macro and the handling of error-terminated bus cycles.

19.5.1 Access evaluation macro

The basic operation of the MPU is performed in the access evaluation macro, a hardware structure replicated in the two-dimensional connection matrix. As shown in the following figure, the access evaluation macro inputs the crossbar bus address phase signals and the contents of a region descriptor (RGD*n*) and performs two major functions:

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- Region hit determination
- Detection of an access protection violation

The following figure shows a functional block diagram.

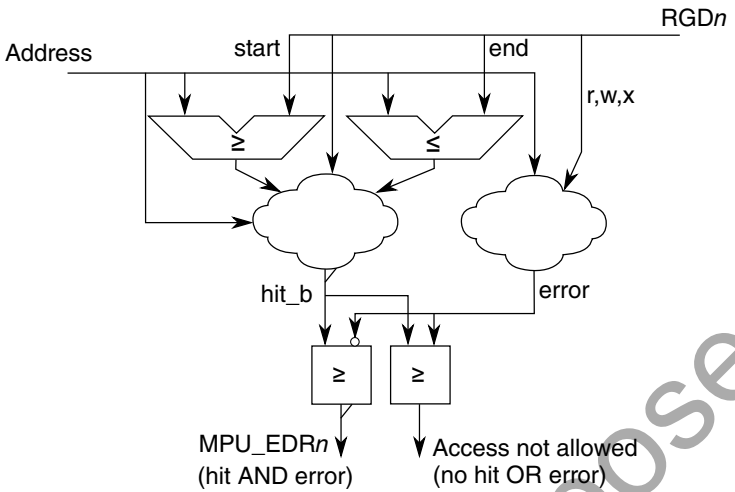


Figure 19-2. MPU access evaluation macro

19.5.1.1 Hit determination

To determine whether the current reference hits in the given region, two magnitude comparators are used with the region's start and end addresses. The boolean equation for this portion of the hit determination is:

```
region_hit = ((addr[31:5] >= RGDn_Word0[SRTADDR]) & (addr[31:5] <= RGDn_Word1[ENDADDR])) &
RGDn_Word3[VLD]
```

where addr is the current reference address, RGD*n*_Word0[SRTADDR] and RGD*n*_Word1[ENDADDR] are the start and end addresses, and RGD*n*_Word3[VLD] is the valid bit.

NOTE

The MPU does not verify that ENDADDR ≥ SRTADDR.

In addition to the comparison of the reference address versus the region descriptor's start and end addresses, the optional process identifier is examined against the region descriptor's PID and PIDMASK fields. A process identifier hit term is formed as follows:

```
pid_hit = ~RGDn_Word2[MxPE] |
          ((current_pid |
          RGDn_Word3[PIDMASK]) == (RGDn_Word3[PID] | RGDn_Word3[PIDMASK]))
```

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

where the current_pid is the selected process identifier from the current bus master, and RGD*n*_Word3[PID] and RGD*n*_Word3[PIDMASK] are the process identifier fields from region descriptor *n*. For bus masters that do not output a process identifier, the MPU forces the pid_hit term to assert.

19.5.1.2 Privilege violation determination

While the access evaluation macro is determining region hit, the logic is also evaluating if the current access is allowed by the permissions defined in the region descriptor. Using the master and supervisor/user mode signals, a set of effective permissions is generated from the appropriate fields in the region descriptor. The protection violation logic then evaluates the access against the effective permissions using the specification shown below.

Table 19-4. Protection violation definition

Description	MxUM			Protection violation?
	r	w	x	
Instruction fetch read	—	—	0	Yes, no execute permission
	—	—	1	No, access is allowed
Data read	0	—	—	Yes, no read permission
	1	—	—	No, access is allowed
Data write	—	0	—	Yes, no write permission
	—	1	—	No, access is allowed

19.5.2 Putting it all together and error terminations

For each slave port monitored, the MPU performs a reduction-AND of all the individual terms from each access evaluation macro. This expression then terminates the bus cycle with an error and reports a protection error for three conditions:

- If the access does not hit in any region descriptor, a protection error is reported.
- If the access hits in a single region descriptor and that region signals a protection violation, a protection error is reported.
- If the access hits in multiple (overlapping) regions and all regions signal protection violations, a protection error is reported.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

As shown in the third condition, granting permission is a higher priority than denying access for overlapping regions. This approach is more flexible to system software in region descriptor assignments. For an example of the use of overlapping region descriptors, see Application information.

19.5.3 Power management

Disabling the MPU by clearing CESR[VLD] minimizes power dissipation. To minimize the power dissipation of an enabled MPU, invalidate unused region descriptors by clearing the associated RGDn_Word3[VLD] bits.

19.6 Initialization information

At system startup, load the appropriate number of region descriptors, including setting RGD*n*_Word3[VLD]. Setting CESR[VLD] enables the module.

If the system requires that all the loaded region descriptors be enabled simultaneously, first ensure that the entire MPU is disabled (CESR[VLD]=0).

Note

A region descriptor must be set to allow access to the MPU registers if further changes are needed.

19.7 Application information

In an operational system, interfacing with the MPU is generally classified into the following activities:

- Creating a new memory region—Load the appropriate region descriptor into an available RGD*n*, using four sequential 32-bit writes. The hardware assists in the maintenance of the valid bit, so if this approach is followed, there are no coherency issues with the multi-cycle descriptor writes. (Clearing RGD*n*_Word3[VLD] deletes/removes an existing memory region.)
- Altering only access privileges—To not affect the valid bit, write to the alternate version of the access control word (RGDAAC*n*), so there are no coherency issues involved with the update. When the write completes, the memory region's access rights switch instantaneously to the new value.

For Assessment Purposes Only

- Changing a region's start and end addresses—Write a minimum of three words to the region descriptor (RGD*n*_Word{0,1,3}). Word 0 and 1 redefine the start and end addresses, respectively. Word 3 re-enables the region descriptor valid bit. In most situations, all four words of the region descriptor are rewritten.
- Accessing the MPU—Allocate a region descriptor to restrict MPU access to supervisor mode from a specific master.
- Detecting an access error—The current bus cycle is terminated with an error response and EAR*n* and EDR*n* capture information on the faulting reference. The error-terminated bus cycle typically initiates an error response in the originating bus master. For example, a processor core may respond with a bus error exception, while a data movement bus master may respond with an error interrupt. The processor can retrieve the captured error address and detail information simply by reading E{A,D}R*n*. CESR[SPERR] signals which error registers contain captured fault data.
- Overlapping region descriptors—Applying overlapping regions often reduces the number of descriptors required for a given set of access controls. In the overlapping memory space, the protection rights of the corresponding region descriptors are logically summed together (the boolean OR operator).

  The following dual-core system example contains four bus masters:

  - The two processors: CP0, CP1
  - Two DMA engines: DMA1, a traditional data movement engine transferring data between RAM and peripherals and DMA2, a second engine transferring data to/from the RAM only

  Consider the following region descriptor assignments:

ssment Purposes Only

Table 19-5. Overlapping region descriptor example

Region description	RGDn		CP0	CP1	DMA1	DMA2	
CP0 code	0		rwx	r--	—	—	Flash
CP1 code	1		r--	rwx	—	—	
CP0 data & stack	2		rw-	—	—	—	RAM
CP0 → CP1 shared data	2	3	r--	r--	—	—	
CP1 → CP0 shared data	4						
CP1 data & stack	4		—	rw-	—	—	
Shared DMA data	5		rw-	rw-	rw	rw	
MPU	6		rw-	rw-	—	—	Peripheral space
Peripherals	7		rw-	rw-	rw	—	

Freescale Confidential Proprietary - Non-Disclosure Agreement required

In this example, there are eight descriptors used to span nine regions in the three main spaces of the system memory map: flash, RAM, and peripheral space. Each region indicates the specific permissions for each of the four bus masters and this definition provides an appropriate set of shared, private and executable memory spaces.

Of particular interest are the two overlapping spaces: region descriptors 2 & 3 and 3 & 4.

The space defined by RGD2 with no overlap is a private data and stack area that provides read/write access to CP0 only. The overlapping space between RGD2 and RGD3 defines a shared data space for passing data from CP0 to CP1 and the access controls are defined by the logical OR of the two region descriptors. Thus, CP0 has (rw- | r--) = (rw-) permissions, while CP1 has (--- | r--) = (r--) permission in this space. Both DMA engines are excluded from this shared processor data region. The overlapping spaces between RGD3 and RGD4 defines another shared data space, this one for passing data from CP1 to CP0. For this overlapping space, CP0 has (r-- | ---) = (r--) permission, while CP1 has (rw- | r--) = (rw-) permission. The non-overlapped space of RGD4 defines a private data and stack area for CP1 only.

The space defined by RGD5 is a shared data region, accessible by all four bus masters. Finally, the slave peripheral space mapped onto the IPS bus is partitioned into two regions:

- One containing the MPU's programming model accessible only to the two processor cores
- The remaining peripheral region accessible to both processors and the traditional DMA1 master

This example shows one possible application of the capabilities of the MPU in a typical system.

For Assessment Purposes Only

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 20
AIPS-Lite

20.1 Chip-specific AIPS-Lite information

20.1.1 Instantiation Information

This device contains one peripheral bridge. The peripheral access protection is supported by AIPS. The MPU will not cover peripheral access protection.

20.1.2 Memory maps

The peripheral bridge is used to access the registers of most of the modules on this device. See S32K1xx_memory_map.xlsx attached to Reference Manual for the memory slot assignment.

AIPS_PACR0 – PACR31 refer to on platform peripherals with corresponding AIPS Peripheral bridge slot numbers from 0 -31 . AIPS_OPACR0 – OPACR95 refer to off platform peripherals with corresponding AIPS Peripheral bridge slot numbers from 32 -127.

20.1.2.1 Register reset values

The following table shows chip-specific reset values for AIPS registers:

Table 20-1. Register reset values

Register	Reset values
MPRA	7770_0000h
PACRA	5400_0000h
PACRB	4400_0400h
PACRC	0000_0000h

Table continues on the next page...

r Assessment Purposes Only

Table 20-1. Register reset values (continued)

Register	Reset values
PACRD	4400_0000h
OPACRA	4400_4444h
OPACRB	0004_4440h
OPACRC	0440_0044h
OPACRD	4444_0400h
OPACRE	4000_0040h
OPACRF	4444_4400h
OPACRG	0040_0000h
OPACRH	0040_0000h
OPACRI	0404_4440h
OPACRJ	0044_4400h
OPACRK	0004_0000h
OPACRL	0000_0444h
OPACRM	0000_0000h
OPACRN	0000_0000h
OPACRO	0000_0000h
OPACRP	0000_0000h

For Assessment P

20.2 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.

20.2.1 Features

Key features of the peripheral bridge are:

- Supports peripheral slots with 8-, 16-, and 32-bit datapath width
- Programming model provides memory protection functionality

Freescale Confidential Proprietary - Non-Disclosure Agreement required

20.2.2 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.

20.3 Memory map/register definition

The 32-bit peripheral bridge registers can be accessed only in supervisor mode by trusted bus masters. Additionally, these registers must be read from or written to only by a 32-bit aligned access. The peripheral bridge registers are mapped into the Peripheral Access Control Register A PACRA[PACR0] address space.

20.3.1 AIPS Register Descriptions

20.3.1.1 AIPS Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	Master Privilege A (MPRA)	32	RW	70000000h
20h	Peripheral Access Control (PACRA)	32	RW	00000004h
24h	Peripheral Access Control (PACRB)	32	RW	00000004h
2Ch	Peripheral Access Control (PACRD)	32	RW	00000004h
40h	Off-Platform Peripheral Access Control (OPACRA)	32	RW	00000004h
44h	Off-Platform Peripheral Access Control (OPACRB)	32	RW	00000004h
48h	Off-Platform Peripheral Access Control (OPACRC)	32	RW	00000004h
4Ch	Off-Platform Peripheral Access Control (OPACRD)	32	RW	00000004h
50h	Off-Platform Peripheral Access Control (OPACRE)	32	RW	00000004h
54h	Off-Platform Peripheral Access Control (OPACRF)	32	RW	00000004h
58h	Off-Platform Peripheral Access Control (OPACRG)	32	RW	00000004h
5Ch	Off-Platform Peripheral Access Control (OPACRH)	32	RW	00000004h
60h	Off-Platform Peripheral Access Control (OPACRI)	32	RW	00000004h

Table continues on the next page...

ment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
64h	Off-Platform Peripheral Access Control (OPACRJ)	32	RW	00000004h
68h	Off-Platform Peripheral Access Control (OPACRK)	32	RW	00000004h
6Ch	Off-Platform Peripheral Access Control (OPACRL)	32	RW	00000004h

20.3.1.2 Master Privilege A (MPRA)

20.3.1.2.1 Address

Register	Offset
MPRA	0h

20.3.1.2.2 Function

The MPRA specifies identical 4-bit fields defining the access-privilege level associated with a bus master to various peripherals on the chip. The register provides one field per bus master.

A register field that maps to an unimplemented master or peripheral behaves as read-only-zero.

Each master is assigned a logical ID from 0 to 15. See the master logical ID assignment table in the chip-specific AIPS information.

20.3.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MTR0	MTW 0	MPL0	0	MTR1	MTW 1	MPL1	0	MTR2	MTW 2	MPL2	Reserved			
W																
Reset	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

20.3.1.2.4 Fields

Field	Function
31 —	Reserved
30 MTR0	Master 0 Trusted For Read Determines whether the master is trusted for read accesses. 0b - This master is not trusted for read accesses. 1b - This master is trusted for read accesses.
29 MTW0	Master 0 Trusted For Writes Determines whether the master is trusted for write accesses. 0b - This master is not trusted for write accesses. 1b - This master is trusted for write accesses.
28 MPL0	Master 0 Privilege Level Specifies how the privilege level of the master is determined. 0b - Accesses from this master are forced to user-mode. 1b - Accesses from this master are not forced to user-mode.
27 —	Reserved
26 MTR1	Master 1 Trusted for Read Determines whether the master is trusted for read accesses. 0b - This master is not trusted for read accesses. 1b - This master is trusted for read accesses.
25 MTW1	Master 1 Trusted for Writes Determines whether the master is trusted for write accesses. 0b - This master is not trusted for write accesses. 1b - This master is trusted for write accesses.
24 MPL1	Master 1 Privilege Level Specifies how the privilege level of the master is determined. 0b - Accesses from this master are forced to user-mode. 1b - Accesses from this master are not forced to user-mode.
23 —	Reserved
22 MTR2	Master 2 Trusted For Read Determines whether the master is trusted for read accesses. 0b - This master is not trusted for read accesses. 1b - This master is trusted for read accesses.
21 MTW2	Master 2 Trusted For Writes Determines whether the master is trusted for write accesses. 0b - This master is not trusted for write accesses. 1b - This master is trusted for write accesses.
20 MPL2	Master 2 Privilege Level Specifies how the privilege level of the master is determined. 0b - Accesses from this master are forced to user-mode. 1b - Accesses from this master are not forced to user-mode.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
19-16 —	Reserved
15-12 —	Reserved
11-8 —	Reserved
7-4 —	Reserved
3-0 —	Reserved

20.3.1.3 Peripheral Access Control (PACRA/PACRB/PACRD)

20.3.1.3.1 Address

Register	Offset
PACRA	20h
PACRB	24h
PACRD	2Ch

20.3.1.3.2 Function

Each PACR register consists of eight 4-bit PACR fields. Each PACR field defines the access levels for a particular on-platform peripheral. The peripheral assignment to each PACR field is defined by the memory map slot of the peripheral. See the chip-specific AIPS information for the field assignment of a particular peripheral.

Every PACR field to which no peripheral is assigned is reserved. Reads to reserved locations return zeros, and writes are ignored.

The following table shows whether the PACR fields on this chip are implemented or reserved for each PACR register. PACR registers with all fields reserved are not shown.

The register diagram following the table is drawn for the general case and shows all PACRS fields.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Register	[31:28]	[27:24]	[23:20]	[19:16]	[15:12]	[11:8]	[7:4]	[3:0]
PACRA	PACR0	PACR1	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
PACRB	PACR8	PACR9	Reserved	Reserved	Reserved	PACR13	Reserved	Reserved
PACRD	PACR24	PACR25	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

20.3.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	SP0	WP0	TP0	0	SP1	WP1	TP1	0	SP2	WP2	TP2	0	SP3	WP3	TP3
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SP4	WP4	TP4	0	SP5	WP5	TP5	0	SP6	WP6	TP6	0	SP7	WP7	TP7
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

20.3.1.3.4 Fields

Field	Function
31 —	Reserved
30 SP0	Supervisor Protect 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
29 WP0	Write Protect 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
28 TP0	Trusted Protect 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
27 —	Reserved
26 SP1	Supervisor Protect 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
25 WP1	Write Protect 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
24	Trusted Protect

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
TP1	0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
23 —	Reserved
22 SP2	Supervisor Protect 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
21 WP2	Write Protect 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
20 TP2	Trusted Protect 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
19 —	Reserved
18 SP3	Supervisor Protect 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
17 WP3	Write Protect 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
16 TP3	Trusted Protect 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
15 —	Reserved
14 SP4	Supervisor Protect 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
13 WP4	Write Protect 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
12 TP4	Trusted Protect 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
11 —	Reserved
10 SP5	Supervisor Protect 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
9 WP5	Write Protect 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
8 TP5	Trusted Protect 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
7 —	Reserved
6 SP6	Supervisor Protect 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
5 WP6	Write Protect 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
4 TP6	Trusted Protect 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
3 —	Reserved
2 SP7	Supervisor Protect 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
1 WP7	Write Protect 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
0 TP7	Trusted Protect 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.

Assessn

20.3.1.4 Off-Platform Peripheral Access Control (See Address Table)

20.3.1.4.1 Address

Register	Offset
OPACRA	40h
OPACRB	44h
OPACRC	48h
OPACRD	4Ch
OPACRE	50h
OPACRF	54h
OPACRG	58h
OPACRH	5Ch

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Register	Offset
OPACRI	60h
OPACRJ	64h
OPACRK	68h
OPACRL	6Ch

20.3.1.4.2 Function

Each OPACR register consists of eight 4-bit OPACR fields. Each OPACR field defines the access levels for a particular off-platform peripheral. The peripheral assignment to each OPACR field is defined by the memory map slot of the peripheral. See the chip-specific AIPS information for the field assignment of a particular peripheral.

Every OPACR field to which no peripheral is assigned is reserved. Reads to reserved locations return zeros, and writes are ignored.

The following table shows the location of each peripheral slot's OPACR field in the OPACR registers.

The register diagram following the table is drawn for the general case and shows all OPACRS fields.

Register	[31:28]	[27:24]	[23:20]	[19:16]	[15:12]	[11:8]	[7:4]	[3:0]
OPACRE	OPACR0	OPACR1	Reserved	Reserved	OPACR4	OPACR5	OPACR6	OPACR7
OPACRF	Reserved	Reserved	Reserved	OPACR11	OPACR12	OPACR13	OPACR14	Reserved
OPACRG	Reserved	OPACR17	OPACR18	OPACR19	Reserved	OPACR21	OPACR22	OPACR23
OPACRH	OPACR24	OPACR25	OPACR26	OPACR27	OPACR28	OPACR29	Reserved	OPACR31
OPACRI	OPACR32	Reserved	Reserved	Reserved	Reserved	OPACR37	OPACR38	Reserved
OPACRJ	OPACR40	OPACR41	OPACR42	OPACR43	OPACR44	OPACR45	Reserved	Reserved
OPACRK	Reserved	Reserved	OPACR50	Reserved	Reserved	Reserved	Reserved	Reserved
OPACRL	Reserved	Reserved	OPACR58	Reserved	Reserved	Reserved	OPACR62	Reserved
OPACRM	OPACR64	OPACR65	Reserved	OPACR67	OPACR68	OPACR69	OPACR70	OPACR71
OPACRN	Reserved	Reserved	OPACR74	OPACR75	OPACR76	OPACR77	Reserved	Reserved
OPACRO	Reserved	Reserved	Reserved	OPACR83	OPACR84	OPACR85	Reserved	Reserved
OPACRP	Reserved	Reserved	Reserved	Reserved	Reserved	OPACR93	OPACR94	OPACR95

For Internal Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

20.3.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	SP0	WP0	TP0	0	SP1	WP1	TP1	0	SP2	WP2	TP2	0	SP3	WP3	TP3
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SP4	WP4	TP4	0	SP5	WP5	TP5	0	SP6	WP6	TP6	0	SP7	WP7	TP7
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

20.3.1.4.4 Fields

Field	Function
31 —	Reserved
30 SP0	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
29 WP0	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
28 TP0	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
27 —	Reserved
26 SP1	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
25	Write Protect

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
WP1	Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
24 TP1	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
23 —	Reserved
22 SP2	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
21 WP2	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
20 TP2	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
19 —	Reserved
18 SP3	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
17 WP3	Write Protect Determines whether the peripheral allows write accesss. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
16 TP3	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
15 —	Reserved
14 SP4	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
13 WP4	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
12 TP4	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
11 —	Reserved
10 SP5	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
9 WP5	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
8 TP5	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
7 —	Reserved
6 SP6	Supervisor Protect

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
5 WP6	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
4 TP6	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.
3 —	Reserved
2 SP7	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0b - This peripheral does not require supervisor privilege level for accesses. 1b - This peripheral requires supervisor privilege level for accesses.
1 WP7	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0b - This peripheral allows write accesses. 1b - This peripheral is write protected.
0 TP7	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0b - Accesses from an untrusted master are allowed. 1b - Accesses from an untrusted master are not allowed.

For R

20.4 Functional description

The peripheral bridge functions as a bus protocol translator between the crossbar switch and the slave peripheral bus.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

20.4.1 Access support

Aligned and misaligned 32-bit, 16-bit, and byte accesses are supported for 32-bit peripherals. Misaligned accesses are supported to allow memory to be placed on the slave peripheral bus. Peripheral registers must not be misaligned, although no explicit checking is performed by the peripheral bridge. All accesses are performed with a single transfer.

All accesses to the peripheral slots must be sized less than or equal to the designated peripheral slot size. If an access is attempted that is larger than the targeted port, an error response is generated.

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 21
Direct Memory Access Multiplexer (DMAMUX)

21.1 Chip-specific Direct Memory Access Multiplexer (DMAMUX) information

21.1.1 DMA transfers via TRGMUX trigger

The triggers from TRGMUX module can trigger a DMA transfer on the first four DMA channels, for example, the LPIT can trigger DMA via TRGMUX. The assignments are detailed at LPIT/DMA Periodic Trigger Assignments .

21.2 Introduction

21.2.1 Overview

The Direct Memory Access Multiplexer (DMAMUX) routes DMA sources, called slots, to any of the 16 DMA channels. This process is illustrated in the following figure.

For Assessment Purposes Only

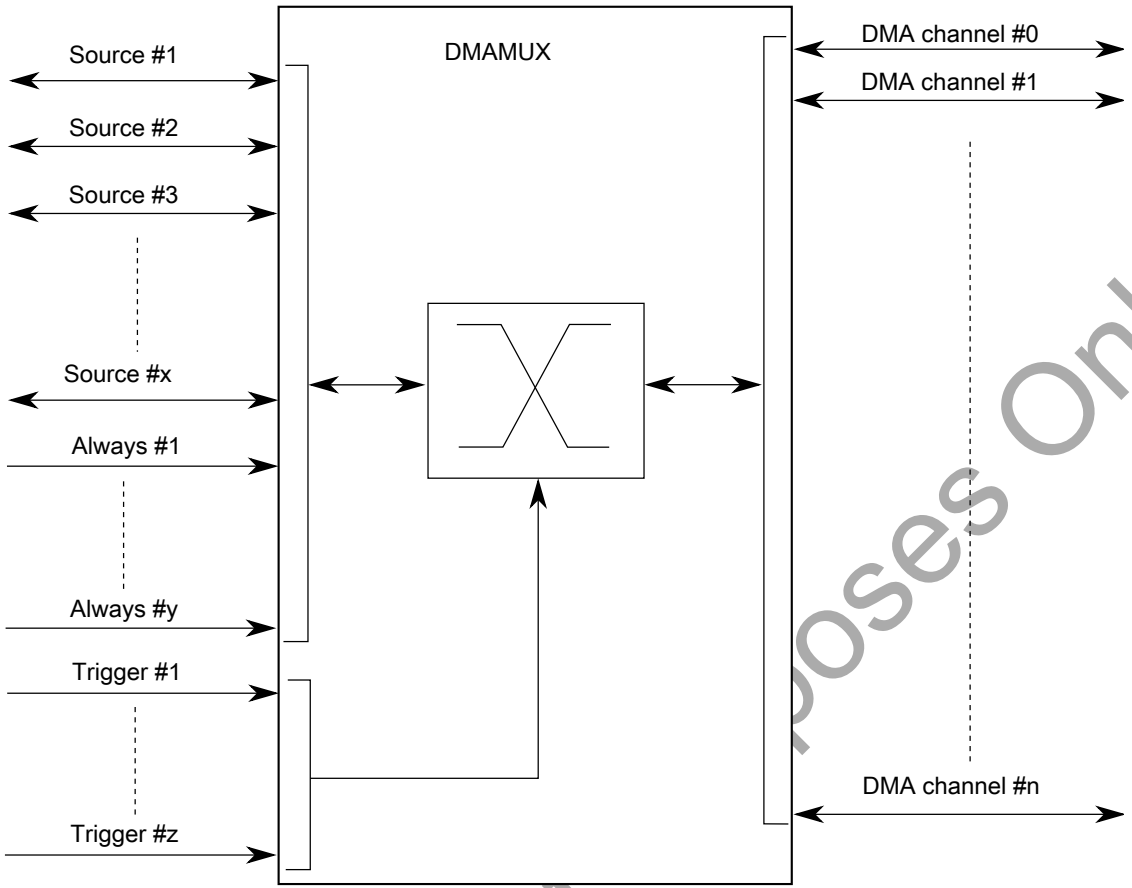


Figure 21-1. DMAMUX block diagram

21.2.2 Features

The DMAMUX module provides these features:

- Up to 61 peripheral slots and up to 2 always-on slots can be routed to 16 channels.
- 16 independently selectable DMA channel routers.
  - The first 4 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.

21.2.3 Modes of operation

The following operating modes are available:

- Disabled mode

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 in the registers of the periodic interrupt timer (PIT). This mode is available only for channels 0–3.

21.3 External signal description

The DMAMUX has no external pins.

21.4 Memory map/register definition

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

21.4.1 DMAMUX Register Descriptions

21.4.1.1 DMAMUX Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h - Fh	Channel Configuration register (CHCFG0 - CHCFG15)	8	RW	00h

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

21.4.1.2 Channel Configuration register (CHCFGa)

21.4.1.2.1 Address

For a = 0 to 15:

Register	Offset
CHCFGa	0h + (a × 1h)

21.4.1.2.2 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].

21.4.1.2.3 Diagram

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

21.4.1.2.4 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

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Field	Function
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.

21.5 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

21.5.1 DMA channels with periodic triggering capability

Besides the normal routing functionality, the first 4 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 the periodic interrupt timer (PIT); as such, the configuration of the periodic triggering interval is done via configuration registers in the PIT. See the section on periodic interrupt timer for more information on this topic.

For Assessment Purposes Only

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.

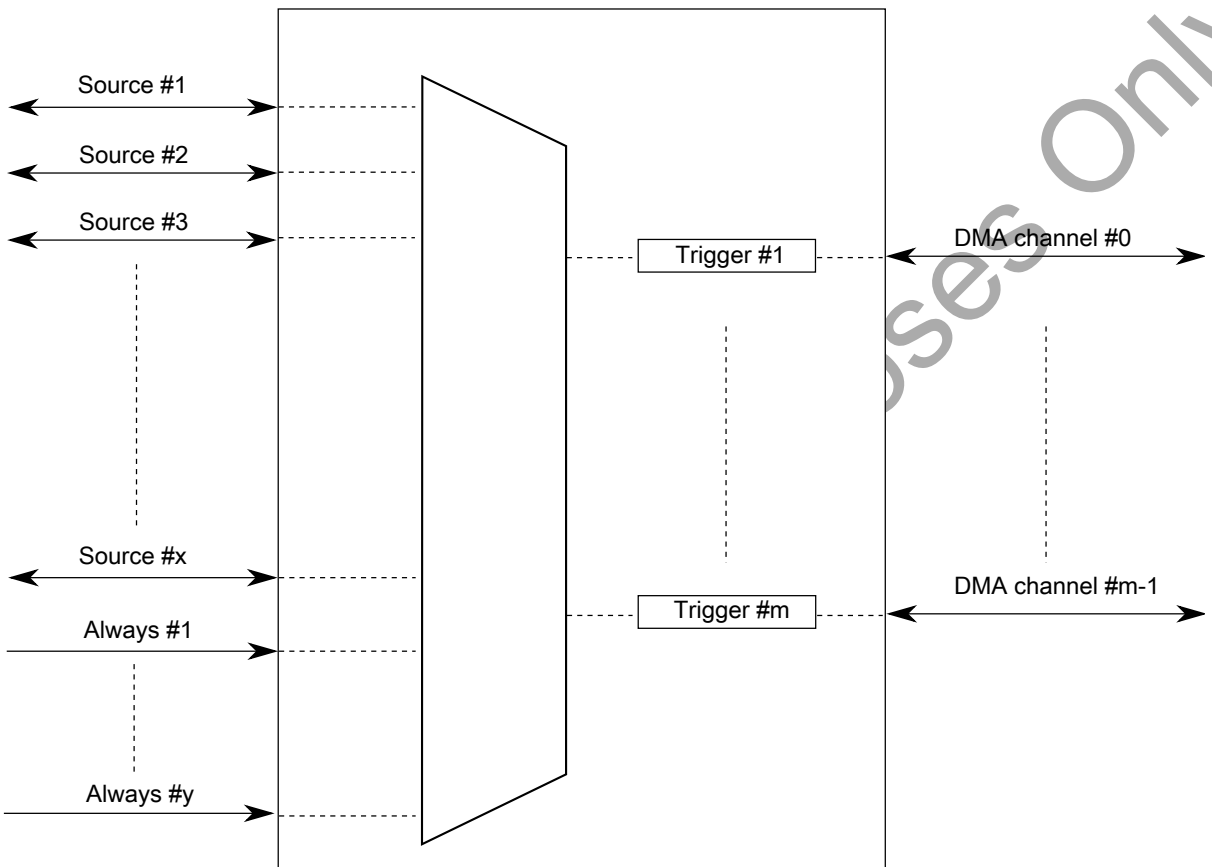


Figure 21-2. DMAMUX triggered channels

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.

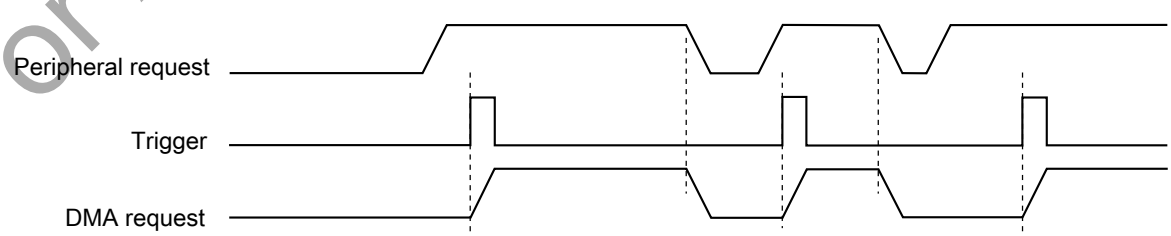


Figure 21-3. DMAMUX channel triggering: normal operation

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

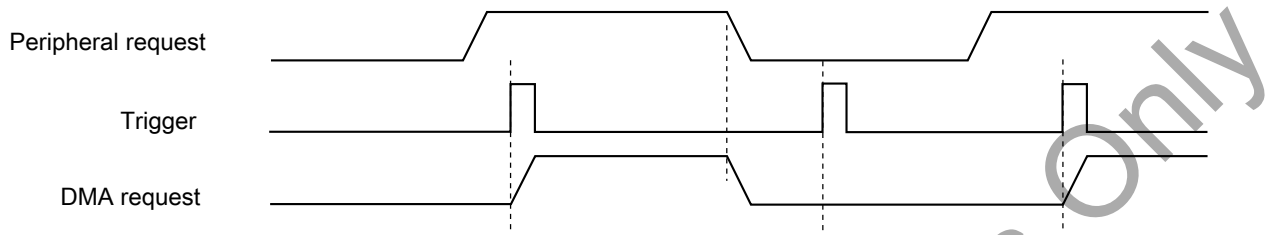


Figure 21-4. DMAMUX channel triggering: ignored trigger

This triggering capability may be used with any peripheral that supports DMA transfers, and is most useful for two types of situations:

- Periodically polling external devices on a particular bus

  As an example, the transmit side of an SPI is assigned to a DMA channel with a trigger, as described above. After it has been set up, the SPI will request DMA transfers, presumably from memory, as long as its transmit buffer is empty. By using a trigger on this channel, the SPI transfers can be automatically performed every 5 µs (as an example). On the receive side of the SPI, the SPI and DMA can be configured to transfer receive data into memory, effectively implementing a method to periodically read data from external devices and transfer the results into memory without processor intervention.

- Using the GPIO ports to drive or sample waveforms

  By configuring the DMA to transfer data to one or more GPIO ports, it is possible to create complex waveforms using tabular data stored in on-chip memory. Conversely, using the DMA to periodically transfer data from one or more GPIO ports, it is possible to sample complex waveforms and store the results in tabular form in on-chip memory.

A more detailed description of the capability of each trigger, including resolution, range of values, and so on, may be found in the periodic interrupt timer section.

For Assessment Purposes Only

21.5.2 DMA channels with no triggering capability

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

21.5.3 Always-enabled DMA sources

In addition to the peripherals that can be used as DMA sources, there are 2 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

21.6 Initialization/application information

This section provides instructions for initializing the DMA channel MUX.

21.6.1 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.

21.6.2 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 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Configure the corresponding timer.
5. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

NOTE

The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with periodic triggering capability:

1. Write 0x00 to CHCFG1.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 4 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+0x0003);
volatile unsigned char *CHCFG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCFG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCFG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCFG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCFG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCFG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCFG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCFG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCFG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCFG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCFG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCFG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);
volatile unsigned char *CHCFG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
```

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

```
volatile unsigned char *CHCFG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCFG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);

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+0x0003);
volatile unsigned char *CHCFG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCFG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCFG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCFG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCFG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCFG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCFG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCFG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCFG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCFG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCFG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCFG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);
volatile unsigned char *CHCFG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCFG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCFG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
```

For Assessment Purposes Only

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

For Assessment Purposes Only

Chapter 22
Enhanced Direct Memory Access (eDMA)

22.1 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 16 channels

22.1.1 eDMA system block diagram

Figure 22-1 illustrates the components of the eDMA system, including the eDMA module ("engine").

For Assessment Purposes Only

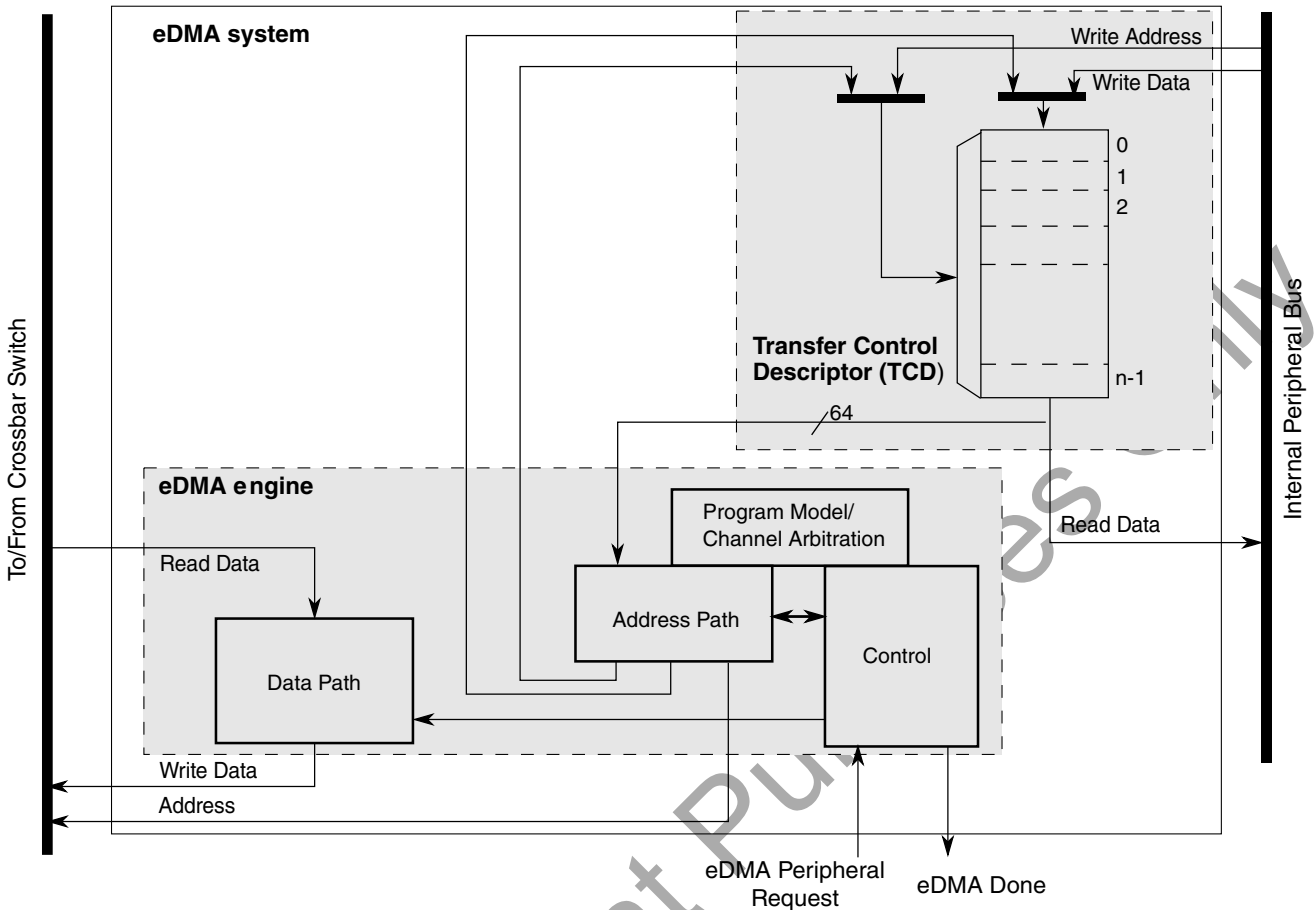


Figure 22-1. eDMA system block diagram

22.1.2 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 22-1. 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...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 22-1. 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 22-2. 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.

22.1.3 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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 16-channel implementation that performs complex data transfers with minimal intervention from a host processor
  - Internal data buffer, used as temporary storage to support 16-byte transfers
  - 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.

For Assessment Purposes Only

22.2 Modes of operation

The eDMA operates in the following modes:

Table 22-3. 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...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 22-3. 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.

22.3 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

22.3.1 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 15. Each TCD*n* definition is presented as 11 registers of 16 or 32 bits.

22.3.2 TCD initialization

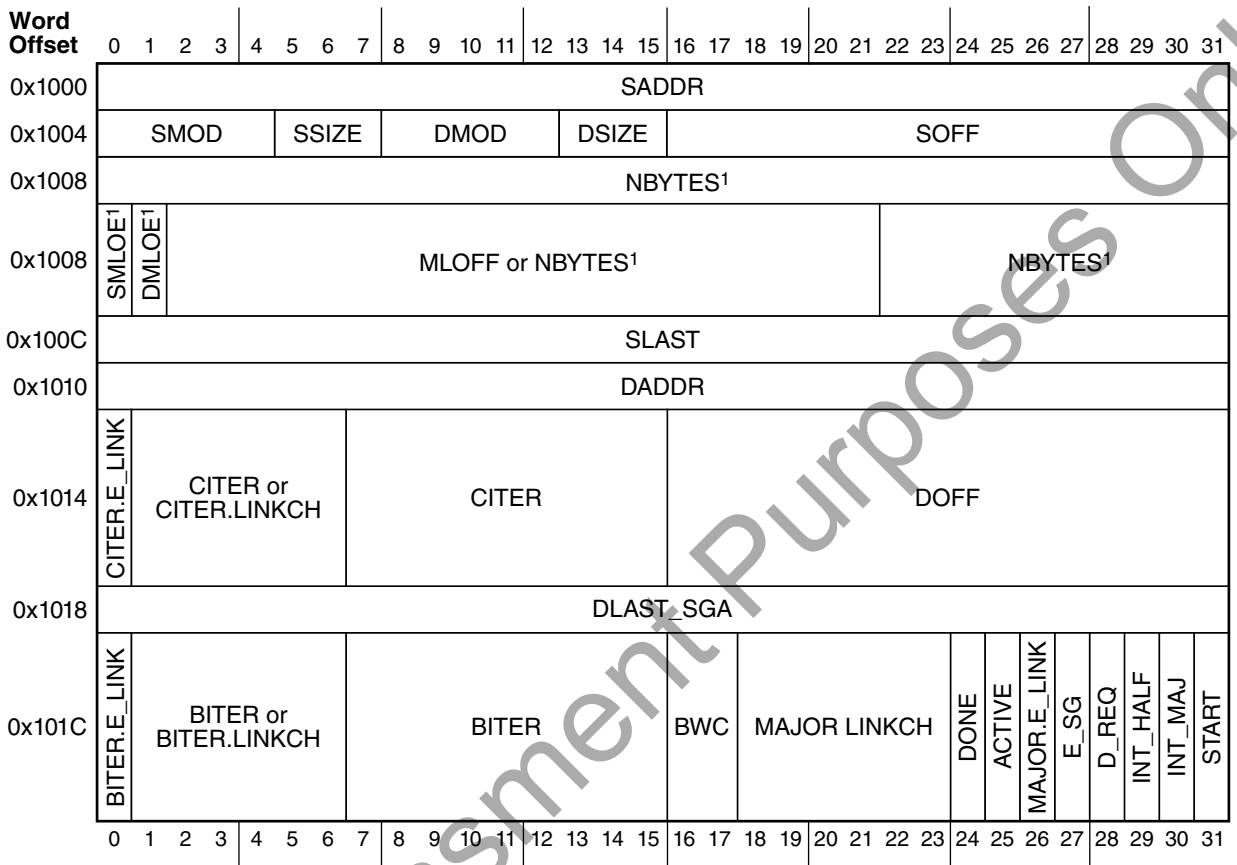
Prior to activating a channel, you must initialize its TCD with the appropriate transfer profile.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.3 TCD structure

DMA Basics: TCD Structure
- One DMA engine has a number of channels to react to DMA requests
- Each channel has its own TCD



[1] The fields implemented in Word 2 depend on whether EDMA_CR[EMLM] bit is set to 0 or 1.

22.3.4 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.

22.3.5 DMA Register Descriptions

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.1 DMA Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	Control (CR)	32	RW	00000000h
4h	Error Status (ES)	32	RO	00000000h
Ch	Enable Request (ERQL)	32	RW	00000000h
14h	Enable Error Interrupt Low (EEIL)	32	RW	00000000h
18h	Clear Enable Error Interrupt (CEEI)	8	WORZ	00h
19h	Set Enable Error Interrupt (SEEI)	8	WORZ	00h
1Ah	Clear Enable Request (CERQ)	8	WORZ	00h
1Bh	Set Enable Request (SERQ)	8	WORZ	00h
1Ch	Clear DONE Status Bit (CDNE)	8	WORZ	00h
1Dh	Set START Bit (SSRT)	8	WORZ	00h
1Eh	Clear Error (CERR)	8	WORZ	00h
1Fh	Clear Interrupt Request (CINT)	8	WORZ	00h
24h	Interrupt Request Low (INTL)	32	W1C	00000000h
2Ch	Error Low (ERRL)	32	W1C	00000000h
34h	Hardware Request Status (HRS)	32	RO	00000000h
44h	Enable Asynchronous Request in Stop (EARS)	32	RW	00000000h
100h - 10Fh	Channel n Priority (DCHPRI3 - DCHPRI18)	8	RW	See description.
1000h	TCD Source Address (TCD0_SADDR)	32	RW	See description.
1004h	TCD Signed Source Address Offset (TCD0_SOFF)	16	RW	See description.
1006h	TCD Transfer Attributes (TCD0_ATTR)	16	RW	See description.
1008h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD0_NBYTES_MLOFFYES)	32	RW	See description.
1008h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD0_NBYTES_MLOFFNO)	32	RW	See description.
1008h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD0_NBYTES_MLNO)	32	RW	See description.
100Ch	TCD Last Source Address Adjustment (TCD0_SLAST)	32	RW	See description.
1010h	TCD Destination Address (TCD0_DADDR)	32	RW	See description.
1014h	TCD Signed Destination Address Offset (TCD0_DOFF)	16	RW	See description.
1016h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_CITER_ELINKNO)	16	RW	See description.
1016h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_CITER_ELINKYES)	16	RW	See description.
1018h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD0_DLASTSGA)	32	RW	See description.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
101Ch	TCD Control and Status (TCD0_CSR)	16	RW	See description.
101Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_BITER_ELINKNO)	16	RW	See description.
101Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_BITER_ELINKYES)	16	RW	See description.
1020h	TCD Source Address (TCD1_SADDR)	32	RW	See description.
1024h	TCD Signed Source Address Offset (TCD1_SOFF)	16	RW	See description.
1026h	TCD Transfer Attributes (TCD1_ATTR)	16	RW	See description.
1028h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD1_NBYTES_MLNO)	32	RW	See description.
1028h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD1_NBYTES_MLOFFNO)	32	RW	See description.
1028h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD1_NBYTES_MLOFFYES)	32	RW	See description.
102Ch	TCD Last Source Address Adjustment (TCD1_SLAST)	32	RW	See description.
1030h	TCD Destination Address (TCD1_DADDR)	32	RW	See description.
1034h	TCD Signed Destination Address Offset (TCD1_DOFF)	16	RW	See description.
1036h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD1_CITER_ELINKYES)	16	RW	See description.
1036h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD1_CITER_ELINKNO)	16	RW	See description.
1038h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD1_DLASTSGA)	32	RW	See description.
103Ch	TCD Control and Status (TCD1_CSR)	16	RW	See description.
103Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD1_BITER_ELINKYES)	16	RW	See description.
103Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD1_BITER_ELINKNO)	16	RW	See description.
1040h	TCD Source Address (TCD2_SADDR)	32	RW	See description.
1044h	TCD Signed Source Address Offset (TCD2_SOFF)	16	RW	See description.
1046h	TCD Transfer Attributes (TCD2_ATTR)	16	RW	See description.
1048h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD2_NBYTES_MLOFFNO)	32	RW	See description.
1048h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD2_NBYTES_MLOFFYES)	32	RW	See description.

Table continues on the next page...

Review Draft
Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
1048h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD2_NBYTES_MLNO)	32	RW	See description.
104Ch	TCD Last Source Address Adjustment (TCD2_SLAST)	32	RW	See description.
1050h	TCD Destination Address (TCD2_DADDR)	32	RW	See description.
1054h	TCD Signed Destination Address Offset (TCD2_DOFF)	16	RW	See description.
1056h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD2_CITER_ELINKNO)	16	RW	See description.
1056h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD2_CITER_ELINKYES)	16	RW	See description.
1058h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD2_DLASTSGA)	32	RW	See description.
105Ch	TCD Control and Status (TCD2_CSR)	16	RW	See description.
105Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD2_BITER_ELINKYES)	16	RW	See description.
105Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD2_BITER_ELINKNO)	16	RW	See description.
1060h	TCD Source Address (TCD3_SADDR)	32	RW	See description.
1064h	TCD Signed Source Address Offset (TCD3_SOFF)	16	RW	See description.
1066h	TCD Transfer Attributes (TCD3_ATTR)	16	RW	See description.
1068h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD3_NBYTES_MLOFFNO)	32	RW	See description.
1068h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD3_NBYTES_MLOFFYES)	32	RW	See description.
1068h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD3_NBYTES_MLNO)	32	RW	See description.
106Ch	TCD Last Source Address Adjustment (TCD3_SLAST)	32	RW	See description.
1070h	TCD Destination Address (TCD3_DADDR)	32	RW	See description.
1074h	TCD Signed Destination Address Offset (TCD3_DOFF)	16	RW	See description.
1076h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD3_CITER_ELINKNO)	16	RW	See description.
1076h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD3_CITER_ELINKYES)	16	RW	See description.
1078h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD3_DLASTSGA)	32	RW	See description.
107Ch	TCD Control and Status (TCD3_CSR)	16	RW	See description.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
107Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD3_BITER_ELINKYES)	16	RW	See description.
107Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD3_BITER_ELINKNO)	16	RW	See description.
1080h	TCD Source Address (TCD4_SADDR)	32	RW	See description.
1084h	TCD Signed Source Address Offset (TCD4_SOFF)	16	RW	See description.
1086h	TCD Transfer Attributes (TCD4_ATTR)	16	RW	See description.
1088h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD4_NBYTES_MLOFFYES)	32	RW	See description.
1088h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD4_NBYTES_MLNO)	32	RW	See description.
1088h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD4_NBYTES_MLOFFNO)	32	RW	See description.
108Ch	TCD Last Source Address Adjustment (TCD4_SLAST)	32	RW	See description.
1090h	TCD Destination Address (TCD4_DADDR)	32	RW	See description.
1094h	TCD Signed Destination Address Offset (TCD4_DOFF)	16	RW	See description.
1096h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD4_CITER_ELINKNO)	16	RW	See description.
1096h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD4_CITER_ELINKYES)	16	RW	See description.
1098h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD4_DLASTSGA)	32	RW	See description.
109Ch	TCD Control and Status (TCD4_CSR)	16	RW	See description.
109Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD4_BITER_ELINKYES)	16	RW	See description.
109Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD4_BITER_ELINKNO)	16	RW	See description.
10A0h	TCD Source Address (TCD5_SADDR)	32	RW	See description.
10A4h	TCD Signed Source Address Offset (TCD5_SOFF)	16	RW	See description.
10A6h	TCD Transfer Attributes (TCD5_ATTR)	16	RW	See description.
10A8h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD5_NBYTES_MLOFFNO)	32	RW	See description.
10A8h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD5_NBYTES_MLNO)	32	RW	See description.
10A8h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD5_NBYTES_MLOFFYES)	32	RW	See description.

Table continues on the next page...

Review Draft
Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
10ACh	TCD Last Source Address Adjustment (TCD5_SLAST)	32	RW	See description.
10B0h	TCD Destination Address (TCD5_DADDR)	32	RW	See description.
10B4h	TCD Signed Destination Address Offset (TCD5_DOFF)	16	RW	See description.
10B6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD5_CITER_ELINKYES)	16	RW	See description.
10B6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD5_CITER_ELINKNO)	16	RW	See description.
10B8h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD5_DLASTSGA)	32	RW	See description.
10BCh	TCD Control and Status (TCD5_CSR)	16	RW	See description.
10BEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD5_BITER_ELINKNO)	16	RW	See description.
10BEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD5_BITER_ELINKYES)	16	RW	See description.
10C0h	TCD Source Address (TCD6_SADDR)	32	RW	See description.
10C4h	TCD Signed Source Address Offset (TCD6_SOFF)	16	RW	See description.
10C6h	TCD Transfer Attributes (TCD6_ATTR)	16	RW	See description.
10C8h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD6_NBYTES_MLOFFNO)	32	RW	See description.
10C8h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD6_NBYTES_MLOFFYES)	32	RW	See description.
10C8h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD6_NBYTES_MLNO)	32	RW	See description.
10CCh	TCD Last Source Address Adjustment (TCD6_SLAST)	32	RW	See description.
10D0h	TCD Destination Address (TCD6_DADDR)	32	RW	See description.
10D4h	TCD Signed Destination Address Offset (TCD6_DOFF)	16	RW	See description.
10D6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD6_CITER_ELINKYES)	16	RW	See description.
10D6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD6_CITER_ELINKNO)	16	RW	See description.
10D8h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD6_DLASTSGA)	32	RW	See description.
10DCh	TCD Control and Status (TCD6_CSR)	16	RW	See description.
10DEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD6_BITER_ELINKYES)	16	RW	See description.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
10DEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD6_BITER_ELINKNO)	16	RW	See description.
10E0h	TCD Source Address (TCD7_SADDR)	32	RW	See description.
10E4h	TCD Signed Source Address Offset (TCD7_SOFF)	16	RW	See description.
10E6h	TCD Transfer Attributes (TCD7_ATTR)	16	RW	See description.
10E8h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD7_NBYTES_MLOFFYES)	32	RW	See description.
10E8h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD7_NBYTES_MLOFFNO)	32	RW	See description.
10E8h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD7_NBYTES_MLNO)	32	RW	See description.
10ECh	TCD Last Source Address Adjustment (TCD7_SLAST)	32	RW	See description.
10F0h	TCD Destination Address (TCD7_DADDR)	32	RW	See description.
10F4h	TCD Signed Destination Address Offset (TCD7_DOFF)	16	RW	See description.
10F6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD7_CITER_ELINKNO)	16	RW	See description.
10F6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD7_CITER_ELINKYES)	16	RW	See description.
10F8h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD7_DLASTSGA)	32	RW	See description.
10FCh	TCD Control and Status (TCD7_CSR)	16	RW	See description.
10FEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD7_BITER_ELINKNO)	16	RW	See description.
10FEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD7_BITER_ELINKYES)	16	RW	See description.
1100h	TCD Source Address (TCD8_SADDR)	32	RW	See description.
1104h	TCD Signed Source Address Offset (TCD8_SOFF)	16	RW	See description.
1106h	TCD Transfer Attributes (TCD8_ATTR)	16	RW	See description.
1108h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD8_NBYTES_MLOFFNO)	32	RW	See description.
1108h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD8_NBYTES_MLOFFYES)	32	RW	See description.
1108h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD8_NBYTES_MLNO)	32	RW	See description.
110Ch	TCD Last Source Address Adjustment (TCD8_SLAST)	32	RW	See description.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
1110h	TCD Destination Address (TCD8_DADDR)	32	RW	See description.
1114h	TCD Signed Destination Address Offset (TCD8_DOFF)	16	RW	See description.
1116h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD8_CITER_ELINKYES)	16	RW	See description.
1116h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD8_CITER_ELINKNO)	16	RW	See description.
1118h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD8_DLASTSGA)	32	RW	See description.
111Ch	TCD Control and Status (TCD8_CSR)	16	RW	See description.
111Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD8_BITER_ELINKYES)	16	RW	See description.
111Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD8_BITER_ELINKNO)	16	RW	See description.
1120h	TCD Source Address (TCD9_SADDR)	32	RW	See description.
1124h	TCD Signed Source Address Offset (TCD9_SOFF)	16	RW	See description.
1126h	TCD Transfer Attributes (TCD9_ATTR)	16	RW	See description.
1128h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD9_NBYTES_MLOFFNO)	32	RW	See description.
1128h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD9_NBYTES_MLNO)	32	RW	See description.
1128h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD9_NBYTES_MLOFFYES)	32	RW	See description.
112Ch	TCD Last Source Address Adjustment (TCD9_SLAST)	32	RW	See description.
1130h	TCD Destination Address (TCD9_DADDR)	32	RW	See description.
1134h	TCD Signed Destination Address Offset (TCD9_DOFF)	16	RW	See description.
1136h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD9_CITER_ELINKNO)	16	RW	See description.
1136h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD9_CITER_ELINKYES)	16	RW	See description.
1138h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD9_DLASTSGA)	32	RW	See description.
113Ch	TCD Control and Status (TCD9_CSR)	16	RW	See description.
113Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD9_BITER_ELINKYES)	16	RW	See description.
113Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD9_BITER_ELINKNO)	16	RW	See description.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
1140h	TCD Source Address (TCD10_SADDR)	32	RW	See description.
1144h	TCD Signed Source Address Offset (TCD10_SOFF)	16	RW	See description.
1146h	TCD Transfer Attributes (TCD10_ATTR)	16	RW	See description.
1148h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD10_NBYTES_MLOFFNO)	32	RW	See description.
1148h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD10_NBYTES_MLNO)	32	RW	See description.
1148h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD10_NBYTES_MLOFFYES)	32	RW	See description.
114Ch	TCD Last Source Address Adjustment (TCD10_SLAST)	32	RW	See description.
1150h	TCD Destination Address (TCD10_DADDR)	32	RW	See description.
1154h	TCD Signed Destination Address Offset (TCD10_DOFF)	16	RW	See description.
1156h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD10_CITER_ELINKNO)	16	RW	See description.
1156h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD10_CITER_ELINKYES)	16	RW	See description.
1158h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD10_DLASTSGA)	32	RW	See description.
115Ch	TCD Control and Status (TCD10_CSR)	16	RW	See description.
115Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD10_BITER_ELINKYES)	16	RW	See description.
115Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD10_BITER_ELINKNO)	16	RW	See description.
1160h	TCD Source Address (TCD11_SADDR)	32	RW	See description.
1164h	TCD Signed Source Address Offset (TCD11_SOFF)	16	RW	See description.
1166h	TCD Transfer Attributes (TCD11_ATTR)	16	RW	See description.
1168h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD11_NBYTES_MLNO)	32	RW	See description.
1168h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD11_NBYTES_MLOFFNO)	32	RW	See description.
1168h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD11_NBYTES_MLOFFYES)	32	RW	See description.
116Ch	TCD Last Source Address Adjustment (TCD11_SLAST)	32	RW	See description.
1170h	TCD Destination Address (TCD11_DADDR)	32	RW	See description.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
1174h	TCD Signed Destination Address Offset (TCD11_DOFF)	16	RW	See description.
1176h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD11_CITER_ELINKYES)	16	RW	See description.
1176h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD11_CITER_ELINKNO)	16	RW	See description.
1178h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD11_DLASTSGA)	32	RW	See description.
117Ch	TCD Control and Status (TCD11_CSR)	16	RW	See description.
117Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD11_BITER_ELINKNO)	16	RW	See description.
117Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD11_BITER_ELINKYES)	16	RW	See description.
1180h	TCD Source Address (TCD12_SADDR)	32	RW	See description.
1184h	TCD Signed Source Address Offset (TCD12_SOFF)	16	RW	See description.
1186h	TCD Transfer Attributes (TCD12_ATTR)	16	RW	See description.
1188h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD12_NBYTES_MLOFFYES)	32	RW	See description.
1188h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD12_NB YTES_MLNO)	32	RW	See description.
1188h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD12_NBYTES_MLOFFNO)	32	RW	See description.
118Ch	TCD Last Source Address Adjustment (TCD12_SLAST)	32	RW	See description.
1190h	TCD Destination Address (TCD12_DADDR)	32	RW	See description.
1194h	TCD Signed Destination Address Offset (TCD12_DOFF)	16	RW	See description.
1196h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD12_CITER_ELINKNO)	16	RW	See description.
1196h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD12_CITER_ELINKYES)	16	RW	See description.
1198h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD12_DLASTSGA)	32	RW	See description.
119Ch	TCD Control and Status (TCD12_CSR)	16	RW	See description.
119Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD12_BITER_ELINKYES)	16	RW	See description.
119Eh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD12_BITER_ELINKNO)	16	RW	See description.
11A0h	TCD Source Address (TCD13_SADDR)	32	RW	See description.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
11A4h	TCD Signed Source Address Offset (TCD13_SOFF)	16	RW	See description.
11A6h	TCD Transfer Attributes (TCD13_ATTR)	16	RW	See description.
11A8h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD13_NBYTES_MLOFFYES)	32	RW	See description.
11A8h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD13_NBYTES_MLNO)	32	RW	See description.
11A8h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD13_NBYTES_MLOFFNO)	32	RW	See description.
11ACh	TCD Last Source Address Adjustment (TCD13_SLAST)	32	RW	See description.
11B0h	TCD Destination Address (TCD13_DADDR)	32	RW	See description.
11B4h	TCD Signed Destination Address Offset (TCD13_DOFF)	16	RW	See description.
11B6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD13_CITER_ELINKYES)	16	RW	See description.
11B6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD13_CITER_ELINKNO)	16	RW	See description.
11B8h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD13_DLASTSGA)	32	RW	See description.
11BCh	TCD Control and Status (TCD13_CSR)	16	RW	See description.
11BEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD13_BITER_ELINKNO)	16	RW	See description.
11BEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD13_BITER_ELINKYES)	16	RW	See description.
11C0h	TCD Source Address (TCD14_SADDR)	32	RW	See description.
11C4h	TCD Signed Source Address Offset (TCD14_SOFF)	16	RW	See description.
11C6h	TCD Transfer Attributes (TCD14_ATTR)	16	RW	See description.
11C8h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD14_NBYTES_MLOFFNO)	32	RW	See description.
11C8h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD14_NBYTES_MLOFFYES)	32	RW	See description.
11C8h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD14_NBYTES_MLNO)	32	RW	See description.
11CCh	TCD Last Source Address Adjustment (TCD14_SLAST)	32	RW	See description.
11D0h	TCD Destination Address (TCD14_DADDR)	32	RW	See description.
11D4h	TCD Signed Destination Address Offset (TCD14_DOFF)	16	RW	See description.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
11D6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD14_CITER_ELINKYES)	16	RW	See description.
11D6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD14_CITER_ELINKNO)	16	RW	See description.
11D8h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD14_DLASTSGA)	32	RW	See description.
11DCh	TCD Control and Status (TCD14_CSR)	16	RW	See description.
11DEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD14_BITER_ELINKYES)	16	RW	See description.
11DEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD14_BITER_ELINKNO)	16	RW	See description.
11E0h	TCD Source Address (TCD15_SADDR)	32	RW	See description.
11E4h	TCD Signed Source Address Offset (TCD15_SOFF)	16	RW	See description.
11E6h	TCD Transfer Attributes (TCD15_ATTR)	16	RW	See description.
11E8h	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD15_NBYTES_MLOFFYES)	32	RW	See description.
11E8h	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD15_NBYTES_MLOFFNO)	32	RW	See description.
11E8h	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD15_NBYTES_MLNO)	32	RW	See description.
11ECh	TCD Last Source Address Adjustment (TCD15_SLAST)	32	RW	See description.
11F0h	TCD Destination Address (TCD15_DADDR)	32	RW	See description.
11F4h	TCD Signed Destination Address Offset (TCD15_DOFF)	16	RW	See description.
11F6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD15_CITER_ELINKNO)	16	RW	See description.
11F6h	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD15_CITER_ELINKYES)	16	RW	See description.
11F8h	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD15_DLASTSGA)	32	RW	See description.
11FCh	TCD Control and Status (TCD15_CSR)	16	RW	See description.
11FEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD15_BITER_ELINKYES)	16	RW	See description.
11FEh	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD15_BITER_ELINKNO)	16	RW	See description.

22.3.5.2 Control (CR)

22.3.5.2.1 Address

Register	Offset
CR	0h

22.3.5.2.2 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.

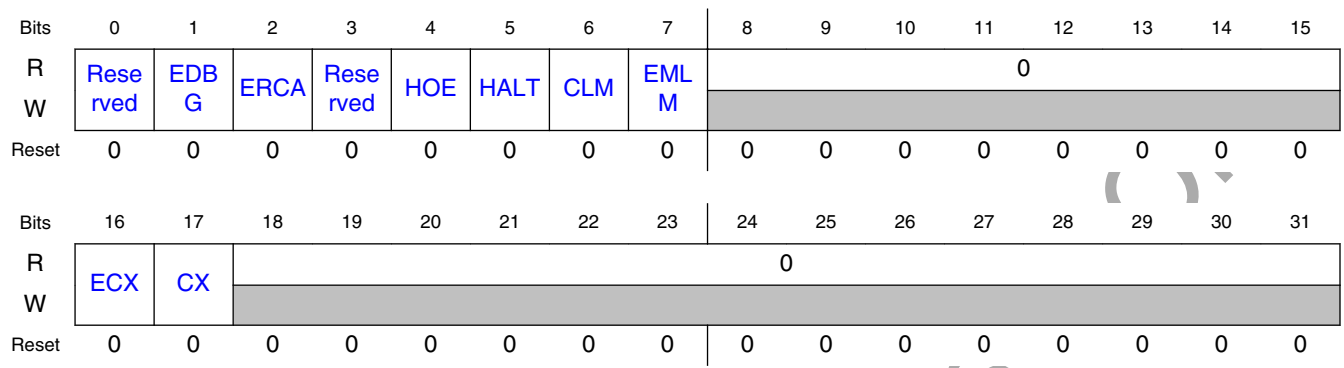
When minor loop mapping is enabled (EMLM is 1), TCDn word2 is redefined. A portion of TCDn word2 is used to specify multiple fields: a source enable bit (SMLOE) to specify the minor loop offset should be applied to the source address (TCDn_SADDR) upon minor loop completion, a destination enable bit (DMLOE) 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

When minor loop mapping is disabled (EMLM is 0), all 32 bits of TCDn word2 are assigned to the NBYTES field.

22.3.5.2.3 Diagram



22.3.5.2.4 Fields

Field	Function
0 —	Reserved Reserved
1 EDBG	Enable Debug 0b - When in debug mode, the DMA continues to operate. 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.
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 .
3 —	Reserved Reserved
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.
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.
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, e.g., 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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	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.
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.
8-15 —	Reserved
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.
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.
18-31 —	Reserved

For Assessment

22.3.5.3 Error Status (ES)

22.3.5.3.1 Address

Register	Offset
ES	4h

22.3.5.3.2 Function

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- A configuration error, that is:

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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 the Error Reporting and Handling section for more details.

22.3.5.3.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	DBE	SBE	SGE	NCE	DOE	DAE	SOE	SAE	ERRCHN				0		CPE	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

22.3.5.3.4 Fields

Field	Function
0 DBE	Destination Bus Error 0b - No destination bus error 1b - The last recorded error was a bus error on a destination write
1 SBE	Source Bus Error 0b - No source bus error 1b - The last recorded error was a bus error on a source read
2 SGE	Scatter/Gather Configuration Error 0b - No scatter/gather configuration error 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.
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]
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].

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
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].
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].
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].
8-11 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.
12-13 —	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.
15 —	Reserved
16 ECX	Transfer Canceled 0b - No canceled transfers 1b - The last recorded entry was a canceled transfer by the error cancel transfer input
17-30 —	Reserved
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.

For As

22.3.5.4 Enable Request (ERQL)

22.3.5.4.1 Address

Register	Offset
ERQL	Ch

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.4.2 Function

The ERQ register provides a bit map for the 16 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 the ERQ.

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.

22.3.5.4.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	ERQ0	ERQ1	ERQ2	ERQ3	ERQ4	ERQ5	ERQ6	ERQ7	ERQ8	ERQ9	ERQ10	ERQ11	ERQ12	ERQ13	ERQ14	ERQ15
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

22.3.5.4.4 Fields

Field	Function
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
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
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
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
4 ERQ4	Enable DMA Request 4 0b - The DMA request signal for the corresponding channel is disabled

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	1b - The DMA request signal for the corresponding channel is enabled
5 ERQ5	Enable DMA Request 5 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
6 ERQ6	Enable DMA Request 6 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
7 ERQ7	Enable DMA Request 7 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
8 ERQ8	Enable DMA Request 8 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
9 ERQ9	Enable DMA Request 9 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
10 ERQ10	Enable DMA Request 10 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
11 ERQ11	Enable DMA Request 11 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
12 ERQ12	Enable DMA Request 12 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
13 ERQ13	Enable DMA Request 13 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
14 ERQ14	Enable DMA Request 14 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
15 ERQ15	Enable DMA Request 15 0b - The DMA request signal for the corresponding channel is disabled 1b - The DMA request signal for the corresponding channel is enabled
16-31 —	Reserved

FC

22.3.5.5 Enable Error Interrupt Low (EEIL)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.5.1 Address

Register	Offset
EEIL	14h

22.3.5.5.2 Function

The EEI register provides a bit map for the 16 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.

22.3.5.5.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	EEI0	EEI1	EEI2	EEI3	EEI4	EEI5	EEI6	EEI7	EEI8	EEI9	EEI10	EEI11	EEI12	EEI13	EEI14	EEI15
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

22.3.5.5.4 Fields

Field	Function
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
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
2 EEI2	Enable Error Interrupt 2 0b - The error signal for corresponding channel does not generate an error interrupt

Table continues on the next page...

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	1b - The assertion of the error signal for corresponding channel generates an error interrupt request
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
4 EEI4	Enable Error Interrupt 4 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
5 EEI5	Enable Error Interrupt 5 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
6 EEI6	Enable Error Interrupt 6 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
7 EEI7	Enable Error Interrupt 7 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
8 EEI8	Enable Error Interrupt 8 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
9 EEI9	Enable Error Interrupt 9 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
10 EEI10	Enable Error Interrupt 10 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
11 EEI11	Enable Error Interrupt 11 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
12 EEI12	Enable Error Interrupt 12 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
13 EEI13	Enable Error Interrupt 13 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
14 EEI14	Enable Error Interrupt 14 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
15 EEI15	Enable Error Interrupt 15 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
16-31 —	Reserved

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.6 Clear Enable Error Interrupt (CEEI)

22.3.5.6.1 Address

Register	Offset
CEEI	18h

22.3.5.6.2 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.

22.3.5.6.3 Diagram

Bits	0	1	2	3	4	5	6	7
R	0						0	0
W	CEEI				0		CAEE	NOP
Reset	0	0	0	0	0	0	0	0

22.3.5.6.4 Fields

Field	Function
0-3 CEEI	Clear Enable Error Interrupt Clears the corresponding bit in EEI
4-5 —	Reserved
6 CAEE	Clear All Enable Error Interrupts 0b - Clear only the EEI bit specified in the CEEI field

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	1b - Clear all bits in EEI
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register

22.3.5.7 Set Enable Error Interrupt (SEEI)

22.3.5.7.1 Address

Register	Offset
SEEI	19h

22.3.5.7.2 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.

22.3.5.7.3 Diagram

Bits	0	1	2	3	4	5	6	7
R	0						0	0
W	SEEI				0		SAEE	NOP
Reset	0	0	0	0	0	0	0	0

22.3.5.7.4 Fields

Field	Function
0-3	Set Enable Error Interrupt

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Assessment Purposes Only

Field	Function
SEEI	Sets the corresponding bit in EEI
4-5 —	Reserved
6 SAEE	Sets All Enable Error Interrupts 0b - Set only the EEI bit specified in the SEEI field. 1b - Sets all bits in EEI
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register

22.3.5.8 Clear Enable Request (CERQ)

22.3.5.8.1 Address

Register	Offset
CERQ	1Ah

22.3.5.8.2 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.

22.3.5.8.3 Diagram

Bits	0	1	2	3	4	5	6	7
R	0						0	0
W	CERQ				0		CAER	NOP
Reset	0	0	0	0	0	0	0	0

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.8.4 Fields

Field	Function
0-3 CERQ	Clear Enable Request Clears the corresponding bit in ERQ.
4-5 —	Reserved
6 CAER	Clear All Enable Requests 0b - Clear only the ERQ bit specified in the CERQ field 1b - Clear all bits in ERQ
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register

22.3.5.9 Set Enable Request (SERQ)

22.3.5.9.1 Address

Register	Offset
SERQ	1Bh

22.3.5.9.2 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.

For Assessment Purpos

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.9.3 Diagram

Bits	0	1	2	3	4	5	6	7
R	0						0	0
W	SERQ				0		SAER	NOP
Reset	0	0	0	0	0	0	0	0

22.3.5.9.4 Fields

Field	Function
0-3 SERQ	Set Enable Request Sets the corresponding bit in ERQ.
4-5 —	Reserved
6 SAER	Set All Enable Requests 0b - Set only the ERQ bit specified in the SERQ field 1b - Set all bits in ERQ
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register

22.3.5.10 Clear DONE Status Bit (CDNE)

22.3.5.10.1 Address

Register	Offset
CDNE	1Ch

22.3.5.10.2 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.

For Assessment Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

22.3.5.10.3 Diagram

Bits	0	1	2	3	4	5	6	7
R	0						0	0
W	CDNE				0		CADN	NOP
Reset	0	0	0	0	0	0	0	0

22.3.5.10.4 Fields

Field	Function
0-3 CDNE	Clear DONE Bit Clears the corresponding bit in TCDn_CSR[DONE]
4-5 —	Reserved
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]
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register

22.3.5.11 Set START Bit (SSRT)

22.3.5.11.1 Address

Register	Offset
SSRT	1Dh

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.11.2 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.

22.3.5.11.3 Diagram

Bits	0	1	2	3	4	5	6	7
R	0						0	0
W	SSRT				0		SAST	NOP
Reset	0	0	0	0	0	0	0	0

22.3.5.11.4 Fields

Field	Function
0-3 SSRT	Set START Bit Sets the corresponding bit in TCDn_CSR[START]
4-5 —	Reserved
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]
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register

22.3.5.12 Clear Error (CERR)

For Development Purposes Only

22.3.5.12.1 Address

Register	Offset
CERR	1Eh

22.3.5.12.2 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.

22.3.5.12.3 Diagram

Bits	0	1	2	3	4	5	6	7
R	0						0	0
W	CERR				0		CAEI	NOP
Reset	0	0	0	0	0	0	0	0

22.3.5.12.4 Fields

Field	Function
0-3 CERR	Clear Error Indicator Clears the corresponding bit in ERR
4-5 —	Reserved
6 CAEI	Clear All Error Indicators 0b - Clear only the ERR bit specified in the CERR field 1b - Clear all bits in ERR
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register

22.3.5.13 Clear Interrupt Request (CINT)

ment purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.13.1 Address

Register	Offset
CINT	1Fh

22.3.5.13.2 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.

22.3.5.13.3 Diagram

Bits	0	1	2	3	4	5	6	7
R	0						0	0
W	CINT				0		CAIR	NOP
Reset	0	0	0	0	0	0	0	0

22.3.5.13.4 Fields

Field	Function
0-3 CINT	Clear Interrupt Request Clears the corresponding bit in INT
4-5 —	Reserved
6 CAIR	Clear All Interrupt Requests 0b - Clear only the INT bit specified in the CINT field 1b - Clear all bits in INT
7 NOP	No Op enable 0b - Normal operation 1b - No operation, ignore the other bits in this register

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.14 Interrupt Request Low (INTL)

22.3.5.14.1 Address

Register	Offset
INTL	24h

22.3.5.14.2 Function

The INT register provides a bit map for the 16 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.

22.3.5.14.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	INT0	INT1	INT2	INT3	INT4	INT5	INT6	INT7	INT8	INT9	INT10	INT11	INT12	INT13	INT14	INT15
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.14.4 Fields

Field	Function
0 INT0	Interrupt Request 0 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
2 INT2	Interrupt Request 2 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
3 INT3	Interrupt Request 3 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
4 INT4	Interrupt Request 4 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
5 INT5	Interrupt Request 5 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
6 INT6	Interrupt Request 6 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
7 INT7	Interrupt Request 7 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
8 INT8	Interrupt Request 8 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
9 INT9	Interrupt Request 9 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
10 INT10	Interrupt Request 10 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
11 INT11	Interrupt Request 11 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
12 INT12	Interrupt Request 12 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
13 INT13	Interrupt Request 13 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
14 INT14	Interrupt Request 14 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
15 INT15	Interrupt Request 15 0b - The interrupt request for corresponding channel is cleared 1b - The interrupt request for corresponding channel is active
16-31 —	Reserved

22.3.5.15 Error Low (ERRL)

22.3.5.15.1 Address

Register	Offset
ERRL	2Ch

22.3.5.15.2 Function

The ERR provides a bit map for the 16 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, 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. 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.15.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	ERR0	ERR1	ERR2	ERR3	ERR4	ERR5	ERR6	ERR7	ERR8	ERR9	ERR10	ERR11	ERR12	ERR13	ERR14	ERR15
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

22.3.5.15.4 Fields

Field	Function
0 ERR0	Error In Channel 0 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
2 ERR2	Error In Channel 2 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
3 ERR3	Error In Channel 3 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
4 ERR4	Error In Channel 4 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
5 ERR5	Error In Channel 5 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
6 ERR6	Error In Channel 6 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
7 ERR7	Error In Channel 7 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
8 ERR8	Error In Channel 8 0b - An error in this channel has not occurred 1b - An error in this channel has occurred

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
9 ERR9	Error In Channel 9 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
10 ERR10	Error In Channel 10 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
11 ERR11	Error In Channel 11 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
12 ERR12	Error In Channel 12 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
13 ERR13	Error In Channel 13 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
14 ERR14	Error In Channel 14 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
15 ERR15	Error In Channel 15 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
16-31 —	Reserved

For Assessment

22.3.5.16 Hardware Request Status (HRS)

22.3.5.16.1 Address

Register	Offset
HRS	34h

22.3.5.16.2 Function

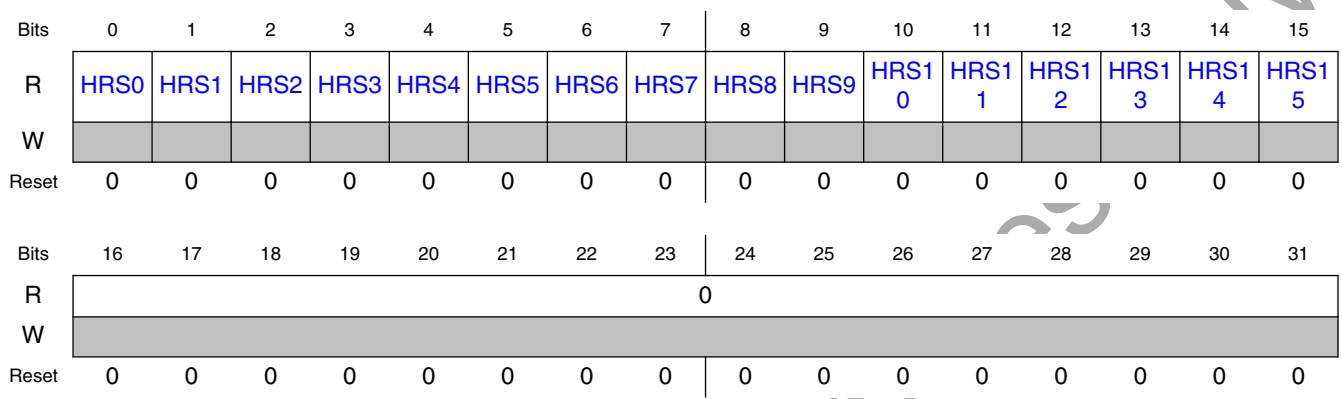
The HRS register provides a bit map for the DMA channels, signaling the presence of a hardware request for each channel. The hardware request status bits reflect the current state of the register and qualified (via the ERQ fields) DMA request signals as seen by the DMA's arbitration logic. This view into the hardware request signals may be used for debug purposes.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

NOTE

These bits reflect the state of the request as seen by the arbitration logic. Therefore, this status is affected by the ERQ bits.

22.3.5.16.3 Diagram



22.3.5.16.4 Fields

Field	Function
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
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
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
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

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
4 HRS4	Hardware Request Status Channel 4 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 4 is not present 1b - A hardware service request for channel 4 is present
5 HRS5	Hardware Request Status Channel 5 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 5 is not present 1b - A hardware service request for channel 5 is present
6 HRS6	Hardware Request Status Channel 6 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 6 is not present 1b - A hardware service request for channel 6 is present
7 HRS7	Hardware Request Status Channel 7 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 7 is not present 1b - A hardware service request for channel 7 is present
8 HRS8	Hardware Request Status Channel 8 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 8 is not present 1b - A hardware service request for channel 8 is present
9 HRS9	Hardware Request Status Channel 9 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 9 is not present 1b - A hardware service request for channel 9 is present
10 HRS10	Hardware Request Status Channel 10 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 10 is not present 1b - A hardware service request for channel 10 is present
11 HRS11	Hardware Request Status Channel 11 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 11 is not present 1b - A hardware service request for channel 11 is present
12	Hardware Request Status Channel 12

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
HRS12	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 12 is not present 1b - A hardware service request for channel 12 is present
13 HRS13	Hardware Request Status Channel 13 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 13 is not present 1b - A hardware service request for channel 13 is present
14 HRS14	Hardware Request Status Channel 14 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 14 is not present 1b - A hardware service request for channel 14 is present
15 HRS15	Hardware Request Status Channel 15 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 15 is not present 1b - A hardware service request for channel 15 is present
16-31 —	Reserved

22.3.5.17 Enable Asynchronous Request in Stop (EARS)

22.3.5.17.1 Address

Register	Offset
EARS	44h

22.3.5.17.2 Function

The EARS register is used to enable or disable the DMA requests in Enable Request (ERQL) by AND'ing the bits of these two registers.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.17.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	EDREQ_0	EDREQ_1	EDREQ_2	EDREQ_3	EDREQ_4	EDREQ_5	EDREQ_6	EDREQ_7	EDREQ_8	EDREQ_9	EDREQ_10	EDREQ_11	EDREQ_12	EDREQ_13	EDREQ_14	EDREQ_15
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

22.3.5.17.4 Fields

Field	Function
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.
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.
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.
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.
4 EDREQ_4	Enable asynchronous DMA request in stop mode for channel 4 0b - Disable asynchronous DMA request for channel 4. 1b - Enable asynchronous DMA request for channel 4.
5 EDREQ_5	Enable asynchronous DMA request in stop mode for channel 5 0b - Disable asynchronous DMA request for channel 5. 1b - Enable asynchronous DMA request for channel 5.
6 EDREQ_6	Enable asynchronous DMA request in stop mode for channel 6 0b - Disable asynchronous DMA request for channel 6. 1b - Enable asynchronous DMA request for channel 6.
7 EDREQ_7	Enable asynchronous DMA request in stop mode for channel 7 0b - Disable asynchronous DMA request for channel 7. 1b - Enable asynchronous DMA request for channel 7.
8 EDREQ_8	Enable asynchronous DMA request in stop mode for channel 8 0b - Disable asynchronous DMA request for channel 8. 1b - Enable asynchronous DMA request for channel 8.
9	Enable asynchronous DMA request in stop mode for channel 9

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
EDREQ_9	0b - Disable asynchronous DMA request for channel 9. 1b - Enable asynchronous DMA request for channel 9.
10 EDREQ_10	Enable asynchronous DMA request in stop mode for channel 10 0b - Disable asynchronous DMA request for channel 10. 1b - Enable asynchronous DMA request for channel 10.
11 EDREQ_11	Enable asynchronous DMA request in stop mode for channel 11 0b - Disable asynchronous DMA request for channel 11. 1b - Enable asynchronous DMA request for channel 11.
12 EDREQ_12	Enable asynchronous DMA request in stop mode for channel 12 0b - Disable asynchronous DMA request for channel 12. 1b - Enable asynchronous DMA request for channel 12.
13 EDREQ_13	Enable asynchronous DMA request in stop mode for channel 13 0b - Disable asynchronous DMA request for channel 13. 1b - Enable asynchronous DMA request for channel 13.
14 EDREQ_14	Enable asynchronous DMA request in stop mode for channel 14 0b - Disable asynchronous DMA request for channel 14. 1b - Enable asynchronous DMA request for channel 14.
15 EDREQ_15	Enable asynchronous DMA request in stop mode for channel 15 0b - Disable asynchronous DMA request for channel 15. 1b - Enable asynchronous DMA request for channel 15.
16-31 —	Reserved

22.3.5.18 Channel n Priority (DCHPRIa)

22.3.5.18.1 Address

For i = 0 to 15 (a = 3 to 18):

Register	Offset
DCHPRIa	100h + (i × 1h)

22.3.5.18.2 Function

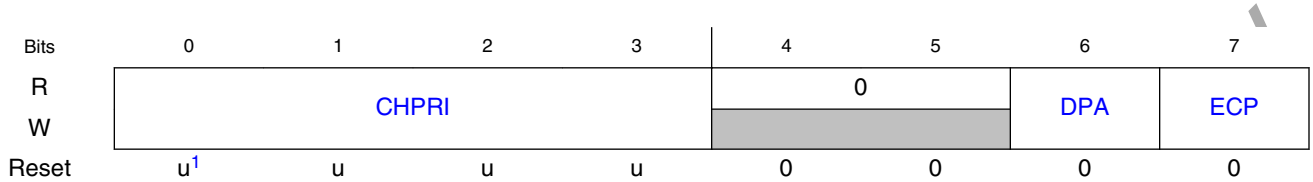
When fixed-priority channel arbitration is enabled (CR[ERCA] = 0), the contents of these registers define the unique priorities associated with each channel. The channel priorities are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 15.

22.3.5.18.3 Diagram



1. See bit field description.

22.3.5.18.4 Fields

Field	Function
0-3 CHPRI	Channel n Arbitration Priority Channel priority when fixed-priority arbitration is enabled **NOTE:** Reset value for the channel priority field, CHPRI, is equal to the corresponding channel number for each priority register, that is, DCHPRI15[CHPRI] = 0b1111.
4-5 —	Reserved
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.
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.

22.3.5.19 TCD Source Address (TCDa_SADDR)

22.3.5.19.1 Address

For a = 0 to 15:

Register	Offset
TCDa_SADDR	1000h + (a × 20h)

For Assessment Purposes Only

22.3.5.19.2 Diagram

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

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

22.3.5.19.3 Fields

Field	Function
0-31 SADDR	Source Address Memory address pointing to the source data.

22.3.5.20 TCD Signed Source Address Offset (TCDa_SOFF)

22.3.5.20.1 Address

For a = 0 to 15:

Register	Offset
TCDa_SOFF	1004h + (a × 20h)

22.3.5.20.2 Diagram

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.20.3 Fields

Field	Function
0-15 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.

22.3.5.21 TCD Transfer Attributes (TCDa_ATTR)

22.3.5.21.1 Address

For a = 0 to 15:

Register	Offset
TCDa_ATTR	1006h + (a × 20h)

22.3.5.21.2 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	DSIZE			DMOD					SSIZE			SMOD				
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

22.3.5.21.3 Fields

Field	Function
0-2 DSIZE	Destination data transfer size See the SSIZE definition
3-7 DMOD	Destination Address Modulo See the SMOD definition
8-10 SSIZE	Source data transfer size **NOTE:** Using a Reserved value causes a configuration error. 000b - 8-bit 001b - 16-bit 010b - 32-bit 011b - 64-bit 100b - Reserved 101b - Reserved 110b - Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	111b - Reserved
11-15 SMOD	Source Address Modulo 00000b - Source address modulo feature is disabled

22.3.5.22 TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCDa_NBYTES_MLNO)

22.3.5.22.1 Address

For a = 0 to 15:

Register	Offset
TCDa_NBYTES_MLNO	1008h + (a × 20h)

22.3.5.22.2 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.

22.3.5.22.3 Diagram

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

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

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.22.4 Fields

Field	Function
0-31 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.

22.3.5.23 TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCDa_NBYTES_MLOFFNO)

22.3.5.23.1 Address

For a = 0 to 15:

Register	Offset
TCDa_NBYTES_MLOFFNO	1008h + (a × 20h)

22.3.5.23.2 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.

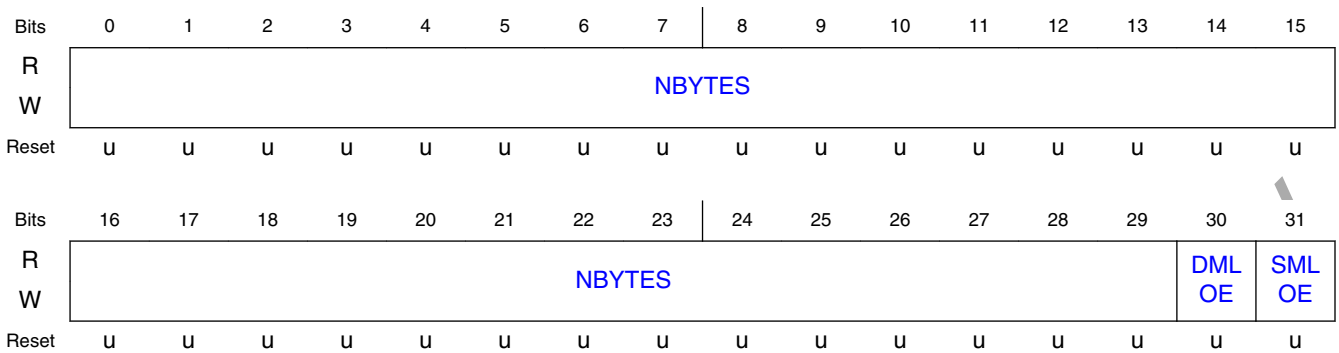
TCD word 2 is defined as follows if:

- 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.

For Assessment Purposes

22.3.5.23.3 Diagram



22.3.5.23.4 Fields

Field	Function
0-29 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.
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
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

22.3.5.24 TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCDa_NBYTES_MLOFFYES)

22.3.5.24.1 Address

For a = 0 to 15:

Register	Offset
TCDa_NBYTES_MLOFFYES	1008h + (a × 20h)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.24.2 Function

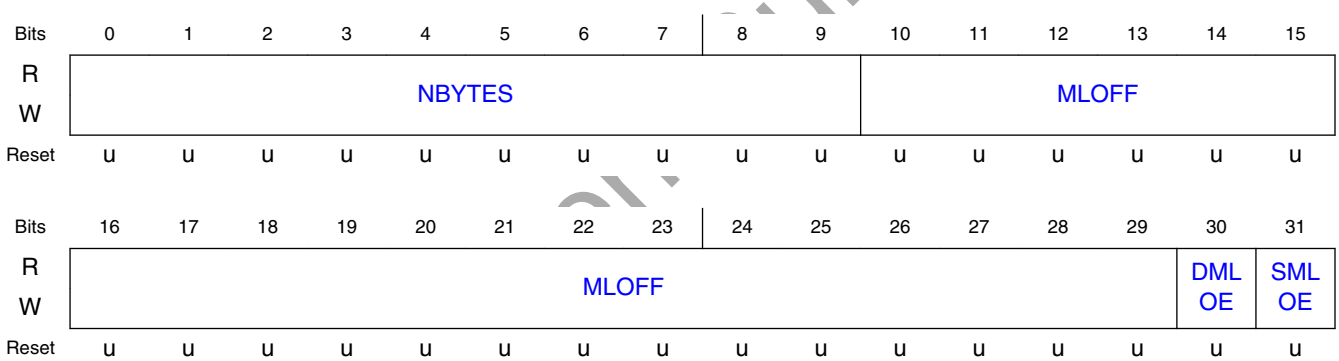
One of three registers (this register, TCD_NBYTES_MLNO, or TCD_NBYTES_MLOFFNO), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

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.

22.3.5.24.3 Diagram



22.3.5.24.4 Fields

Field	Function
0-9 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.
10-29 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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
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
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

22.3.5.25 TCD Last Source Address Adjustment (TCDa_SLAST)

22.3.5.25.1 Address

For a = 0 to 15:

Register	Offset
TCDa_SLAST	100Ch + (a × 20h)

22.3.5.25.2 Diagram

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

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

22.3.5.25.3 Fields

Field	Function
0-31 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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.26 TCD Destination Address (TCDa_DADDR)

22.3.5.26.1 Address

For a = 0 to 15:

Register	Offset
TCDa_DADDR	1010h + (a × 20h)

22.3.5.26.2 Diagram

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

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

22.3.5.26.3 Fields

Field	Function
0-31 DADDR	Destination Address Memory address pointing to the destination data.

22.3.5.27 TCD Signed Destination Address Offset (TCDa_DOFF)

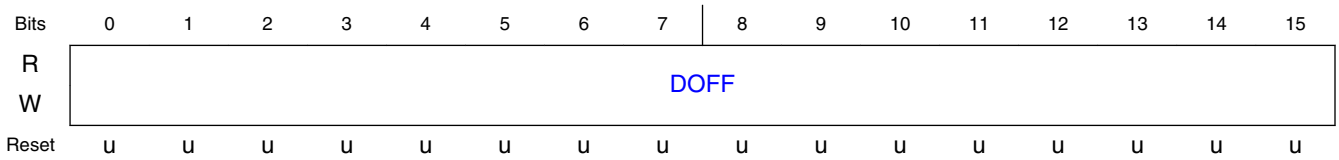
22.3.5.27.1 Address

For a = 0 to 15:

Register	Offset
TCDa_DOFF	1014h + (a × 20h)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.27.2 Diagram



22.3.5.27.3 Fields

Field	Function
0-15 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.

22.3.5.28 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCDa_CITER_ELINKNO)

22.3.5.28.1 Address

For a = 0 to 15:

Register	Offset
TCDa_CITER_ELINKNO	1016h + (a × 20h)

22.3.5.28.2 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) (TCDa_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.

For Assessment Purposes Only

22.3.5.28.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	CITER															ELINK
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

22.3.5.28.4 Fields

Field	Function
0-14 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.
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

22.3.5.29 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCDa_CITER_ELINKYES)

22.3.5.29.1 Address

For a = 0 to 15:

Register	Offset
TCDa_CITER_ELINKYES	1016h + (a × 20h)

For Assessment Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.29.2 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) (TCDa_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.

22.3.5.29.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	CITER_LE									LINKCH						ELINK
W														0		
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

22.3.5.29.4 Fields

Field	Function
0-8 CITER_LE	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.
9-12 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.
13-14 —	Reserved
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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.30 TCD Last Destination Address Adjustment/Scatter Gather Address (TCDa_DLASTSGA)

22.3.5.30.1 Address

For a = 0 to 15:

Register	Offset
TCDa_DLASTSGA	1018h + (a × 20h)

22.3.5.30.2 Diagram

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

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

22.3.5.30.3 Fields

Field	Function
0-31 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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.3.5.31 TCD Control and Status (TCDa_CSR)

22.3.5.31.1 Address

For a = 0 to 15:

Register	Offset
TCDa_CSR	101Ch + (a × 20h)

22.3.5.31.2 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	START	INTMAJOR	INTHALF	DREQ	ESG	MAJORELINK	ACTIVE	DONE	MAJORLINKCH						BWC	
W													0			
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

22.3.5.31.3 Fields

Field	Function
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.
1 INTMAJOR	Enable an interrupt when major iteration count completes. 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.
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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
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.
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.
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.
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.
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.
8-11 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.
12-13 —	Reserved
14-15 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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	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.

22.3.5.32 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCDa_BITER_ELINKNO)

22.3.5.32.1 Address

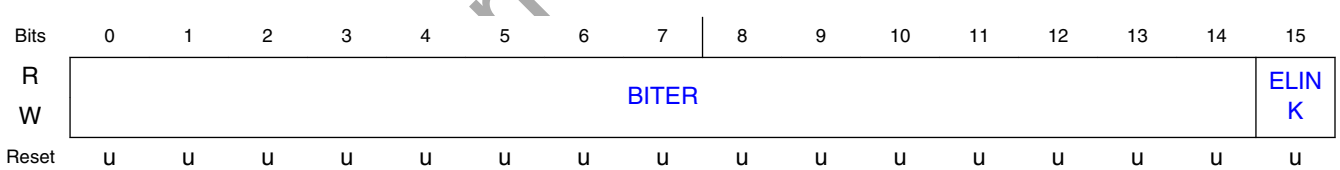
For a = 0 to 15:

Register	Offset
TCDa_BITER_ELINKNO	101Eh + (a × 20h)

22.3.5.32.2 Function

If the TCDn_BITER[ELINK] bit is cleared, the TCDn_BITER register is defined as follows.

22.3.5.32.3 Diagram



22.3.5.32.4 Fields

Field	Function
0-14 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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 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

22.3.5.33 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCDa_BITER_ELINKYES)

22.3.5.33.1 Address

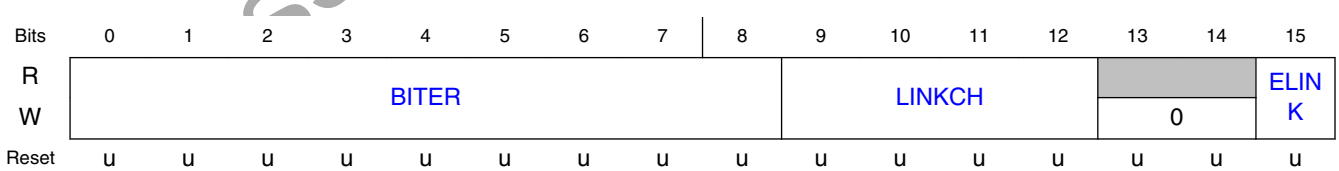
For a = 0 to 15:

Register	Offset
TCDa_BITER_ELINKYES	101Eh + (a × 20h)

22.3.5.33.2 Function

If the TCDn_BITER[ELINK] bit is set, the TCDn_BITER register is defined as follows.

22.3.5.33.3 Diagram



22.3.5.33.4 Fields

Field	Function
0-8 BITER	Starting major iteration count

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	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.
9-12 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.
13-14 —	Reserved
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

For Assessme

22.4 Functional description

The operation of the eDMA is described in the following subsections.

22.4.1 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:

Freescale Confidential Proprietary - Non-Disclosure Agreement required

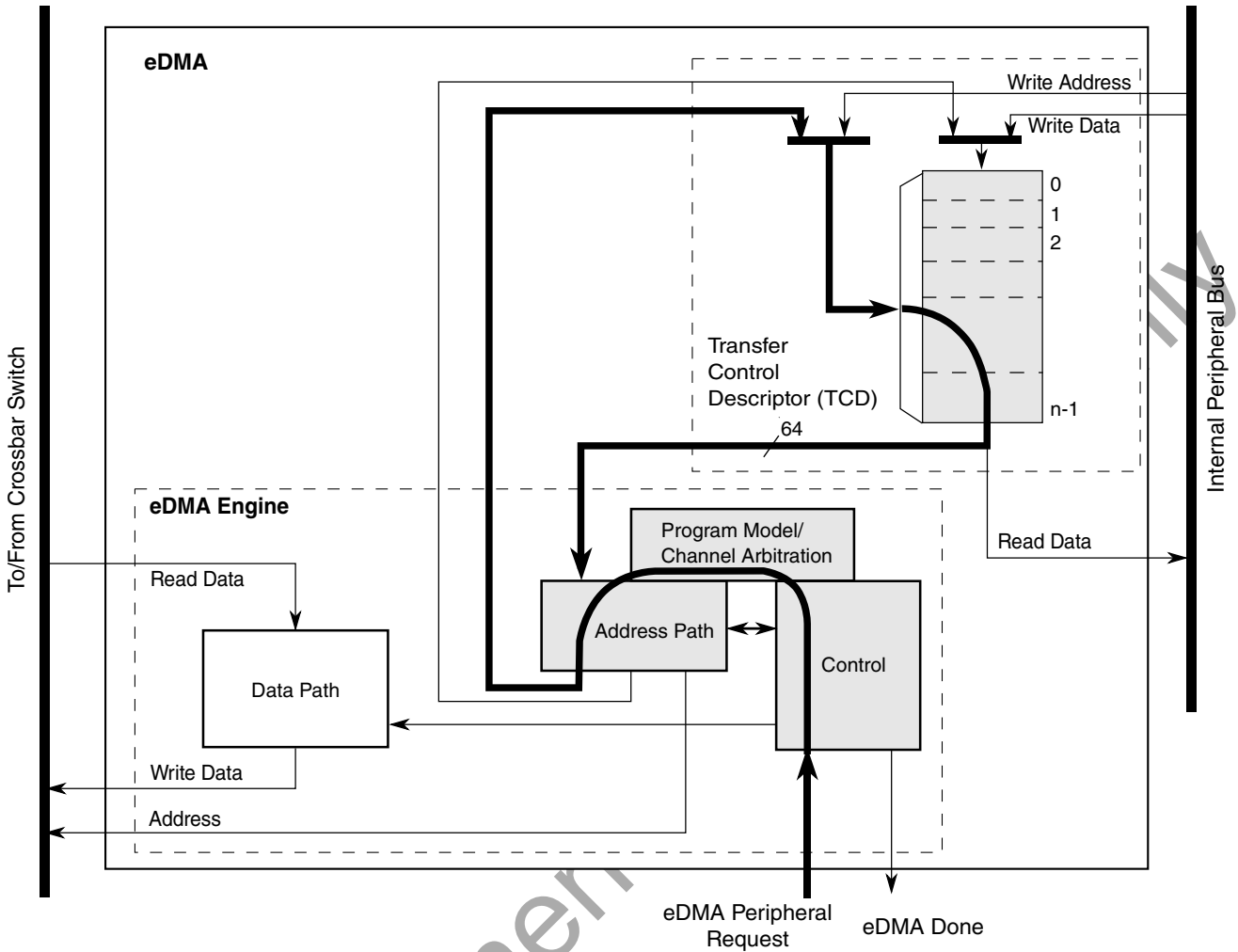


Figure 22-2. eDMA operation, part 1

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:

For Assessment Only

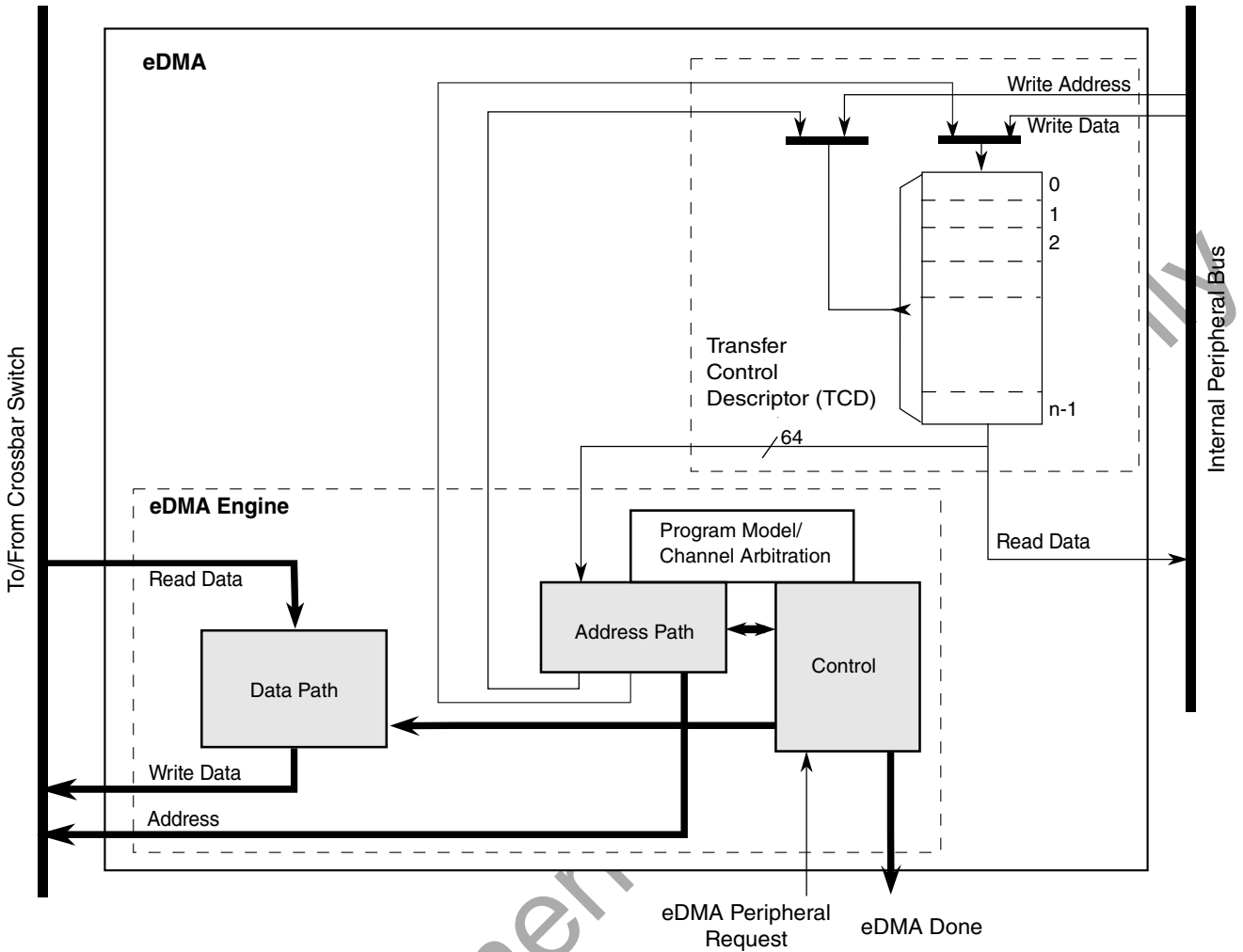


Figure 22-3. eDMA operation, part 2

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.

For Assessment Only

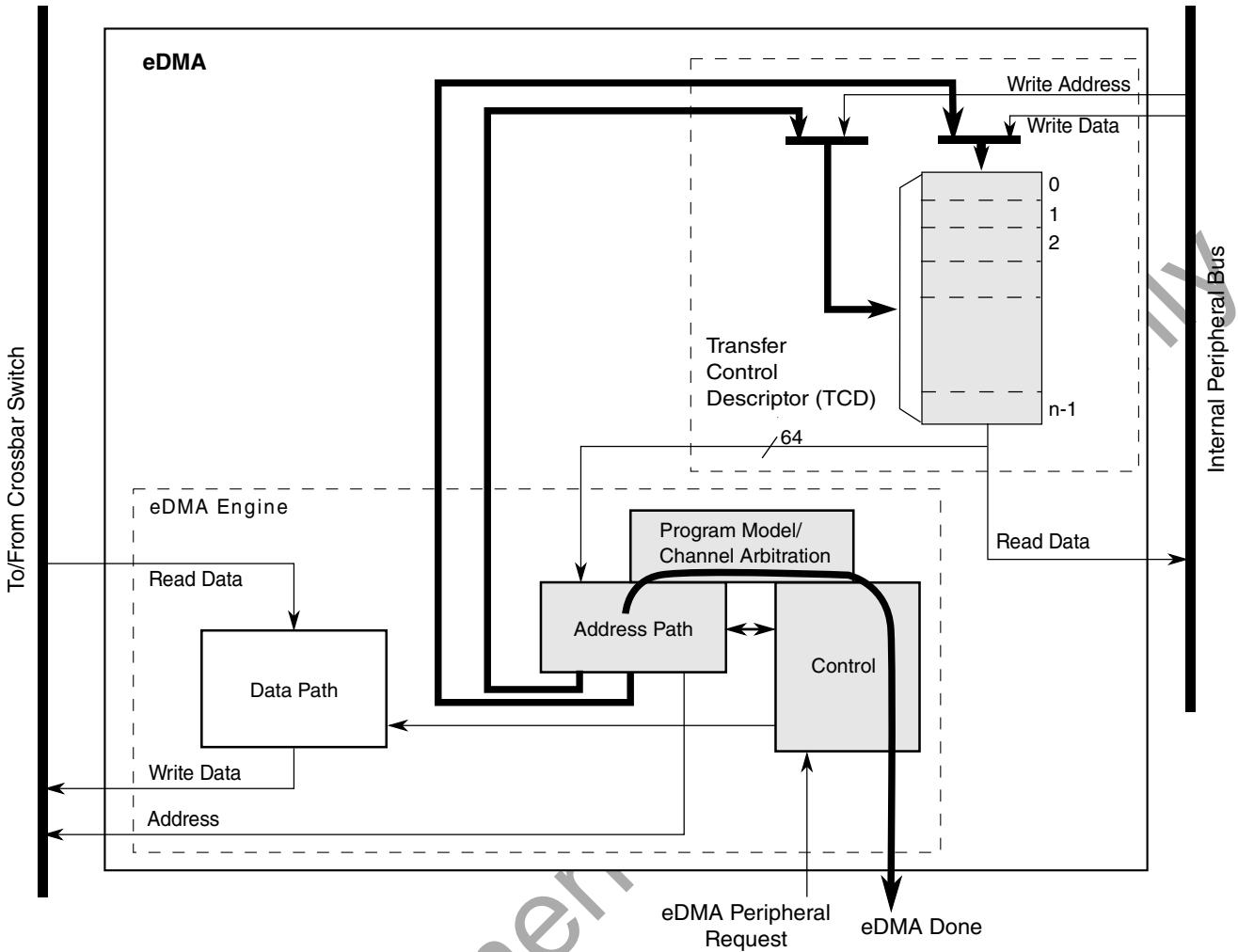


Figure 22-4. eDMA operation, part 3

22.4.2 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.

For Assessment Only

- 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

For Assessment Purposes Only

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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.4.3 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.

22.4.4 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.

22.4.4.1 Peak transfer rates

The peak transfer rates for several different source and destination transfers are shown in the following tables. These tables assume:

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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 22-4. 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
66.7 MHz, 64 bit	266.7	66.6	53.3
83.3 MHz, 64 bit	333.3	83.3	66.7
100.0 MHz, 64 bit	400.0	100.0	80.0
133.3 MHz, 64 bit	533.3	133.3	106.7
150.0 MHz, 64 bit	600.0	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.

22.4.4.2 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 22-5. 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.

For A

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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 22-6. eDMA peak request rate (MReq/sec)

System frequency (MHz)	Request rate with zero wait states	Request rate with wait states
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 22-7. 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)

22.4.4.3 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,

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

22.5 Initialization/application information

The following sections discuss initialization of the eDMA and programming considerations.

22.5.1 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

5. Enable any hardware service requests via the ERQH and ERQL registers.
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.

For Purposes Only

Table 22-8. 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
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).

Freescale Confidential Proprietary - Non-Disclosure Agreement required

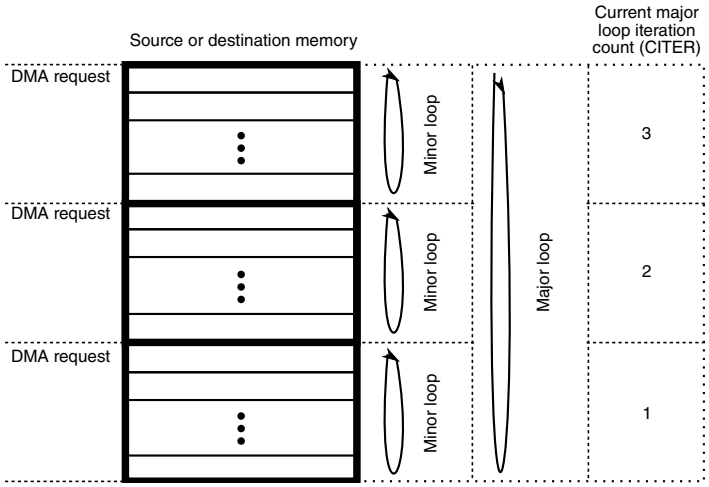


Figure 22-5. Example of multiple loop iterations

The following figure lists the memory array terms and how the TCD settings interrelate.

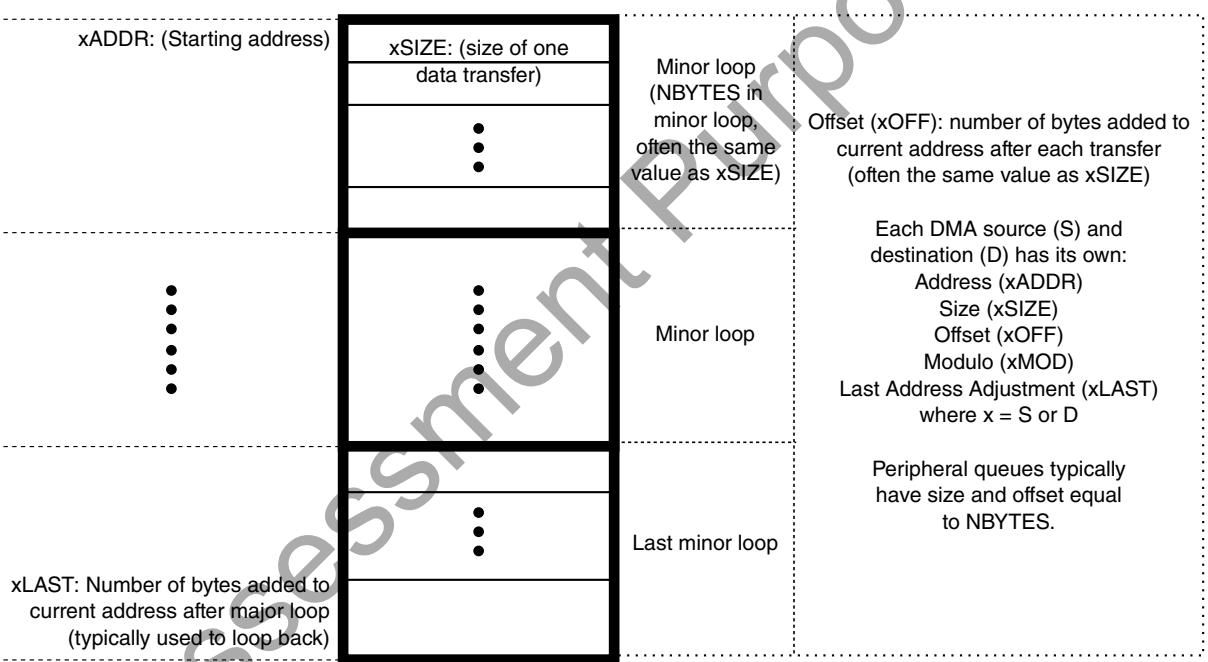


Figure 22-6. Memory array terms

22.5.2 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 Assessment Purposes Only

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.

22.5.3 Arbitration mode considerations

This section discusses arbitration considerations for the eDMA.

22.5.3.1 Fixed channel arbitration

In this mode, the channel service request from the highest priority channel is selected to execute.

22.5.3.2 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.

22.5.4 Performing DMA transfers

This section presents examples on how to perform DMA transfers with the eDMA.

22.5.4.1 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 Assessment Purposes Only

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.

For Assessment Purposes Only

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.

22.5.4.2 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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).

For Assessment Purposes Only

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.

22.5.4.3 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 22-9. Modulo example

Transfer Number	Address
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C
5	0x12345670
6	0x12345674

22.5.5 Monitoring transfer descriptor status

This section discusses how to monitor eDMA status.

22.5.5.1 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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

22.5.5.2 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

For Assessment Purpose

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

22.5.5.3 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.

22.5.6 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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 22-10. 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

22.5.7 Dynamic programming

This section provides recommended methods to change the programming model during channel execution.

22.5.7.1 Dynamically changing the channel priority

The following two options are recommended for dynamically changing channel priority levels:

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

22.5.7.2 Dynamic channel linking

Dynamic channel linking is the process of setting the TCD.major.e_link 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 TCD.major.e_link bit at the same time the eDMA engine is retiring the channel. The TCD.major.e_link 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 TCD.major.e_link bit.
2. Read back the TCD.major.e_link bit.
3. Test the TCD.major.e_link request status:
   - If TCD.major.e_link = 1, the dynamic link attempt was successful.
   - If TCD.major.e_link = 0, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces the TCD.major.e_link 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 TCD.done bit before writing the TCD.major.e_link bit. The TCD.done bit is cleared automatically by the eDMA engine after a channel begins execution.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

22.5.7.3 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 TCD.e_sg bit at the same time the eDMA engine is retiring the channel. The TCD.e_sg 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 major.linkch field and the e_sg 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 TCD.e_sg bits to zero on any writes to a channel's TCD.word7 if that channel's TCD.done bit is set indicating the major loop is complete.

NOTE

The user must clear the TCD.done bit before writing the TCD.major.e_link or TCD.e_sg bits. The TCD.done bit is cleared automatically by the eDMA engine after a channel begins execution.

22.5.7.3.1 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 TCD.major.e_link bit is zero, the TCD.major.linkch field is not used by the eDMA. In this case, the TCD.major.linkch bits may be used for other purposes. This method uses the TCD.major.linkch field as a TCD indentification (ID).

1. When the descriptors are built, write a unique TCD ID in the TCD.major.linkch field for each TCD associated with a channel using dynamic scatter/gather.
2. Write 1b to the TCD.d_req bit.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Should a dynamic scatter/gather attempt fail, setting the TCD.d_req 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 offest value.

3. Write the TCD.dlast_sga field with the scatter/gather address.
4. Write 1b to the TCD.e_sg bit.
5. Read back the 16 bit TCD control/status field.
6. Test the TCD.e_sg request status and TCD.major.linkch value:

   If e_sg = 1b, the dynamic link attempt was successful.

   If e_sg = 0b and the major.linkch (ID) did not change, the attempted dynamic link did not succeed (the channel was already retiring).

   If e_sg = 0b and the major.linkch (ID) changed, the dynamic link attempt was successful (the new TCD's e_sg value cleared the e_sg bit).

22.5.7.3.2 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 indentification (ID).

1. Write 1b to the TCD.d_req bit.

   Should a dynamic scatter/gather attempt fail, setting the d_req 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 offest value.

2. Write theTCD.dlast_sga field with the scatter/gather address.
3. Write 1b to the TCD.e_sg bit.
4. Read back the TCD.e_sg bit.
5. Test the TCD.e_sg request status:

   If e_sg = 1b, the dynamic link attempt was successful.

   If e_sg = 0b, read the 32 bit TCD dlast_sga field.

   If e_sg = 0b and the dlast_sga did not change, the attempted dynamic link did not succeed (the channel was already retiring).

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

If e_sg = 0b and the dlast_sga changed, the dynamic link attempt was successful (the new TCD's e_sg value cleared the e_sg bit).

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 23
Trigger MUX Control (TRGMUX)

23.1 Chip-specific Trigger MUX Control information

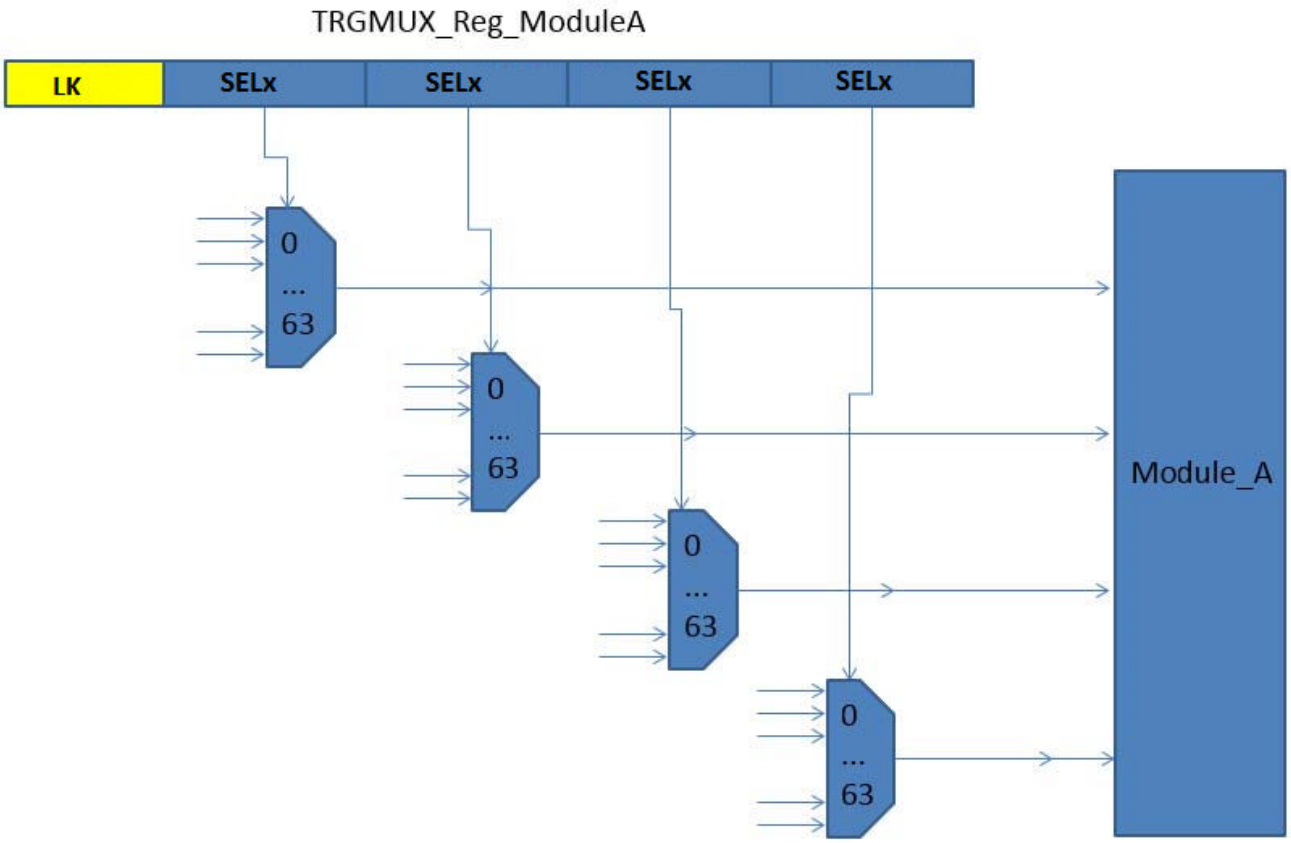
23.1.1 Module Interconnectivity

The TRGMUX introduces an extremely flexible methodology for connecting various trigger sources to multiple pins/peripherals.

With the TRGMUX, each peripheral which accept external triggers will usually have one specific 32-bit trigger control register. Each control register supports up to 4 triggers, and each trigger can be selected from up to 64 inputs.

Following is the main structure of TRGMUX, and take Module_A as an example:

For Assessment Purposes Only



NOTE

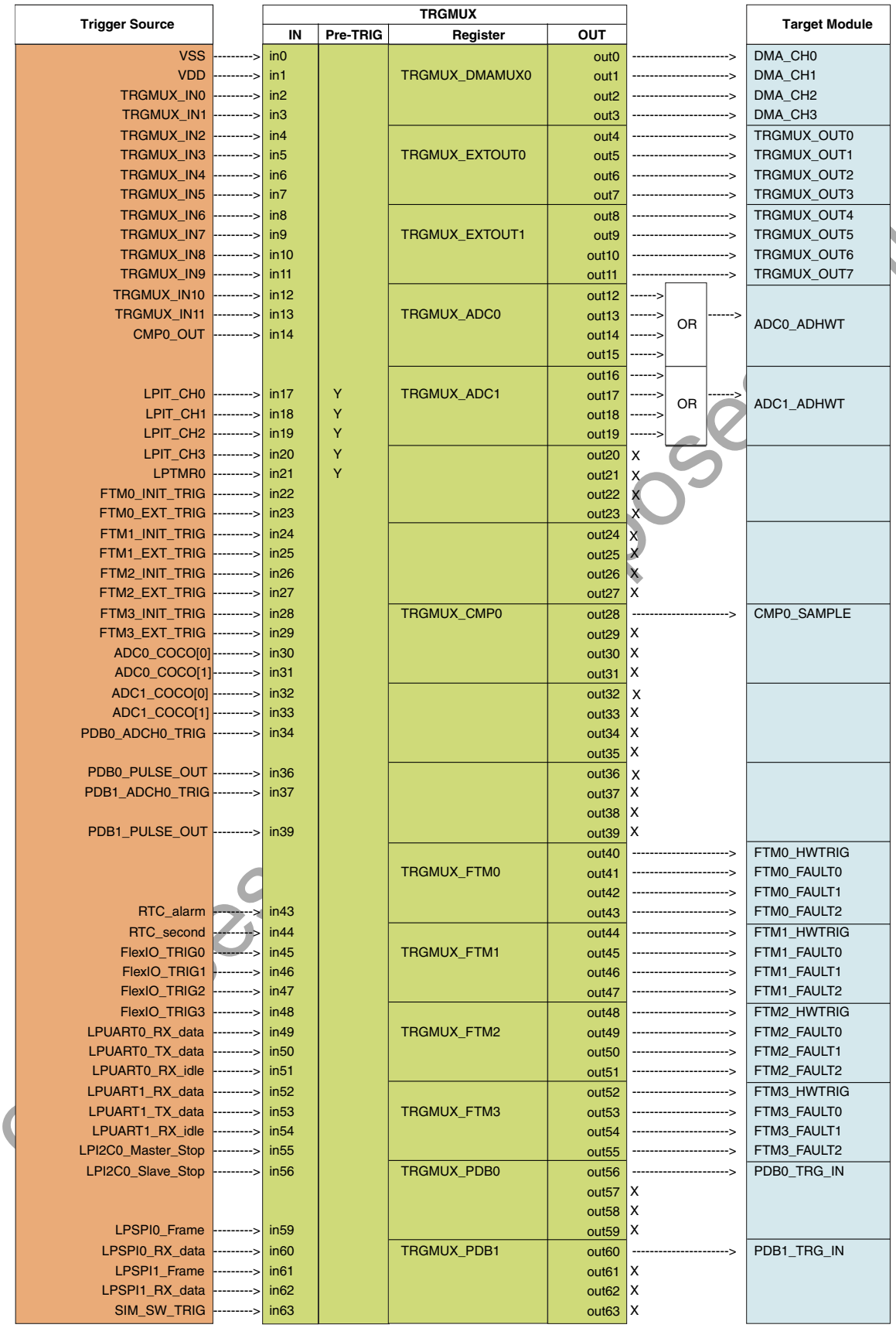
Each TRGMUX control register supports up to 4 trigger channels, but it's not necessary for each module to implement all of the 4 triggers. For those modules (e.g. external output, etc.) which needs more than 4 trigger inputs, multiple control registers are created to support that.

NOTE

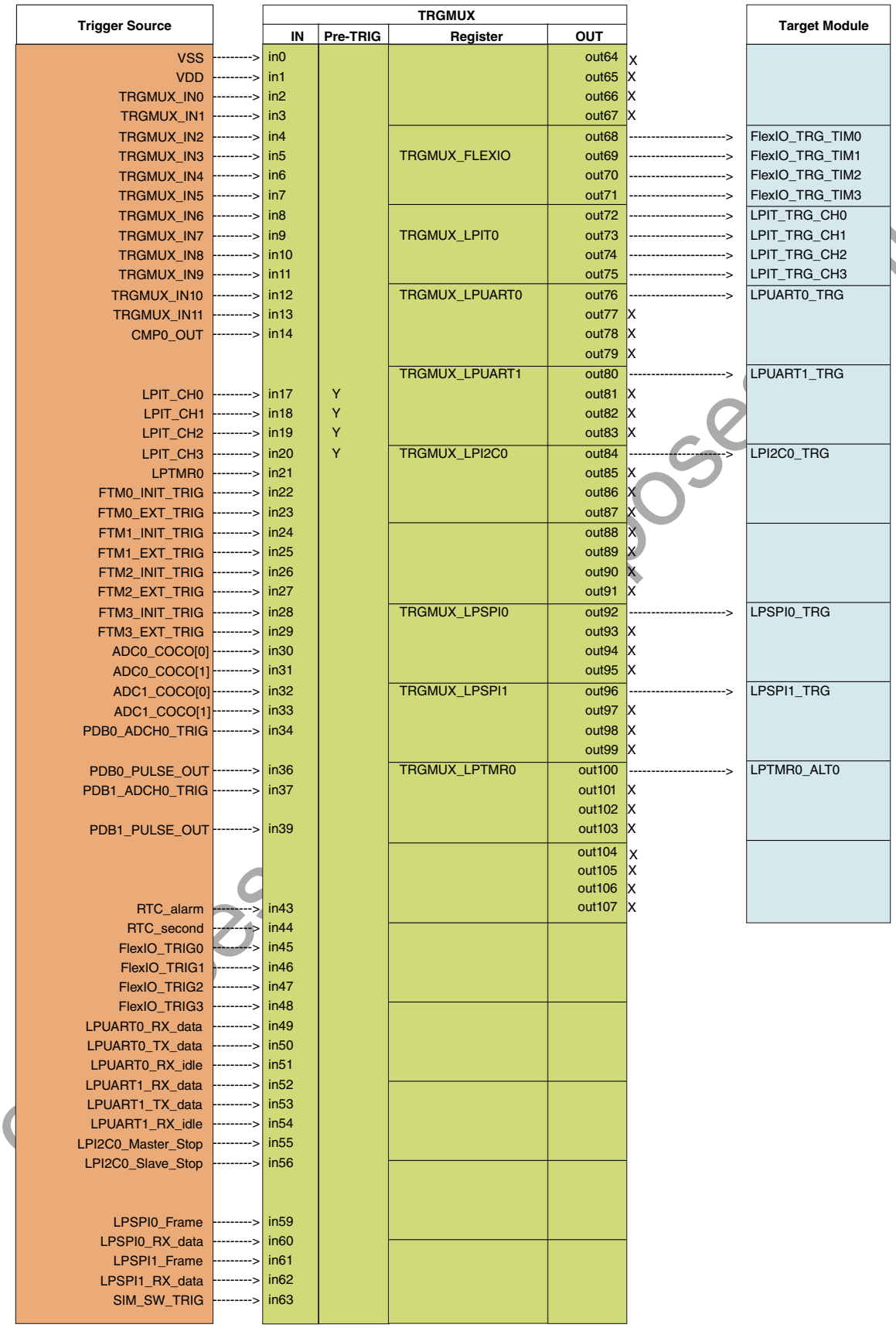
The trigger input and peripheral trigger control are assigned as the following indication.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required



Freescale Confidential Proprietary - Non-Disclosure Agreement required



Trigger Source	TRGMUX IN	TRGMUX Pre-TRIG	TRGMUX Register	TRGMUX OUT	Target Module
VSS	in0			out64 X	
VDD	in1			out65 X	
TRGMUX_IN0	in2			out66 X	
TRGMUX_IN1	in3			out67 X	
TRGMUX_IN2	in4		TRGMUX_FLEXIO	out68	FlexIO_TRG_TIM0
TRGMUX_IN3	in5			out69	FlexIO_TRG_TIM1
TRGMUX_IN4	in6			out70	FlexIO_TRG_TIM2
TRGMUX_IN5	in7			out71	FlexIO_TRG_TIM3
TRGMUX_IN6	in8		TRGMUX_LPIT0	out72	LPIT_TRG_CH0
TRGMUX_IN7	in9			out73	LPIT_TRG_CH1
TRGMUX_IN8	in10			out74	LPIT_TRG_CH2
TRGMUX_IN9	in11			out75	LPIT_TRG_CH3
TRGMUX_IN10	in12		TRGMUX_LPUART0	out76	LPUART0_TRG
TRGMUX_IN11	in13			out77 X	
CMP0_OUT	in14			out78 X	
				out79 X	
			TRGMUX_LPUART1	out80	LPUART1_TRG
LPIT_CH0	in17	Y		out81 X	
LPIT_CH1	in18	Y		out82 X	
LPIT_CH2	in19	Y		out83 X	
LPIT_CH3	in20	Y	TRGMUX_LPI2C0	out84	LPI2C0_TRG
LPTMR0	in21			out85 X	
FTM0_INIT_TRIG	in22			out86 X	
FTM0_EXT_TRIG	in23			out87 X	
FTM1_INIT_TRIG	in24			out88 X	
FTM1_EXT_TRIG	in25			out89 X	
FTM2_INIT_TRIG	in26			out90 X	
FTM2_EXT_TRIG	in27			out91 X	
FTM3_INIT_TRIG	in28		TRGMUX_LPSPI0	out92	LPSPI0_TRG
FTM3_EXT_TRIG	in29			out93 X	
ADC0_COCO[0]	in30			out94 X	
ADC0_COCO[1]	in31			out95 X	
ADC1_COCO[0]	in32		TRGMUX_LPSPI1	out96	LPSPI1_TRG
ADC1_COCO[1]	in33			out97 X	
PDB0_ADCH0_TRIG	in34			out98 X	
				out99 X	
PDB0_PULSE_OUT	in36		TRGMUX_LPTMR0	out100	LPTMR0_ALT0
PDB1_ADCH0_TRIG	in37			out101 X	
				out102 X	
PDB1_PULSE_OUT	in39			out103 X	
				out104 X	
				out105 X	
				out106 X	
RTC_alarm	in43			out107 X	
RTC_second	in44				
FlexIO_TRIG0	in45				
FlexIO_TRIG1	in46				
FlexIO_TRIG2	in47				
FlexIO_TRIG3	in48				
LPUART0_RX_data	in49				
LPUART0_TX_data	in50				
LPUART0_RX_idle	in51				
LPUART1_RX_data	in52				
LPUART1_TX_data	in53				
LPUART1_RX_idle	in54				
LPI2C0_Master_Stop	in55				
LPI2C0_Slave_Stop	in56				
LPSPI0_Frame	in59				
LPSPI0_RX_data	in60				
LPSPI1_Frame	in61				
LPSPI1_RX_data	in62				
SIM_SW_TRIG	in63				

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Freescale Confidential Proprietary - Non-Disclosure Agreement required

NOTE

For each ADC, the four triggers are OR'ed together to provide a flexible trigger scheme for the hardware trigger of each ADC, while the pre-triggers are not ORed. The pre-triggers in TRGMUX can be used with LPIT pre-trigger or software trigger control. There is another PDB pre-trigger scheme existing on this device, which is not through TRGMUX. Please refer to ADC section for details on ADC trigger implementation on this device.

23.2 Introduction

The Trigger MUX module (TRGMUX) allows software to configure the trigger inputs for various peripherals.

23.2.1 Features

The Trigger MUX module allows software to configure the trigger inputs for various peripherals.

- Trigger MUX select

23.2.1.1 Select Bit Fields

Field	Function
5-0	This read/write bit field is used to configure the MUX select for the peripheral trigger inputs.
SEL	000000 - (0x00) Trigger function is disabled.
	000001 - (0x01) Trigger function is disabled.
	000010 - (0x02) Trigger MUX input 0 is selected.
	000011 - (0x03) Trigger MUX input 1 is selected.
	000100 - (0x04) Trigger MUX input 2 is selected.
	000101 - (0x05) Trigger MUX input 3 is selected.
	000110 - (0x06) Trigger MUX input 4 is selected.
	000111 - (0x07) Trigger MUX input 5 is selected.
	001000 - (0x08) Trigger MUX input 6 is selected.
	001001 - (0x09) Trigger MUX input 7 is selected.
	001010 - (0x0A) Trigger MUX input 8 is selected.
	001011 - (0x0B) Trigger MUX input 9 is selected.

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	001100 - (0x0C) Trigger MUX input 10 is selected. 001101 - (0x0D) Trigger MUX input 11 is selected. 001110 - (0x0E) CMP0 Output is selected. 001111 - (0x0F) CMP1 Output is selected. 010000 - (0x10) CMP2 Output is selected. 010001 - (0x11) LPIT Channel 0 is selected. 010010 - (0x12) LPIT Channel 1 is selected. 010011 - (0x13) LPIT Channel 2 is selected. 010100 - (0x14) LPIT Channel 3 is selected. 010101 - (0x15) LPTMR0 Trigger is selected. 010110 - (0x16) FTM0 Internal Trigger is selected. 010111 - (0x17) FTM0 External Trigger is selected. 011000 - (0x18) FTM1 Internal Trigger is selected. 011001 - (0x19) FTM1 External Trigger is selected. 011010 - (0x1A) FTM2 Internal Trigger is selected. 011011 - (0x1B) FTM2 External Trigger is selected. 011100 - (0x1C) FTM3 Internal Trigger is selected. 011101 - (0x1D) FTM3 External Trigger is selected. 011110 - (0x1E) ADC0_COCOA is selected. 011111 - (0x1F) ADC0_COCOB is selected. 100000 - (0x20) ADC1_COCOA is selected. 100001 - (0x21) ADC1_COCOB is selected. 100010 - (0x22) PDB0 Channel 0 Trigger is selected. 100011 - (0x23) PDB0 Channel 1 Trigger is selected. 100100 - (0x24) PDB0 Channel 2 Trigger is selected. 100101 - (0x25) 100110 - (0x26) 100111 - (0x27) 101000 - (0x28) Reserved. 101001 - (0x29) Reserved. 101010 - (0x2A) Reserved. 101011 - (0x2B) RTC Alarm is selected. 101100 - (0x2C) RTC Seconds Counter is selected. 101101 - (0x2D) Flex IO Trigger 0 is selected. 101110 - (0x2E) Flex IO Trigger 1 is selected. 101111 - (0x2F) Flex IO Trigger 2 is selected. 110000 - (0x30) Flex IO Trigger 3 is selected. 110001 - (0x31) LPUART0 RX Data is selected. 110010 - (0x32) LPUART0 TX Data is selected.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	110011 - (0x33) LPUART0 RX Idle is selected. 110100 - (0x34) LPUART1 RX Data is selected. 110101 - (0x35) LPUART1 TX Data is selected. 110110 - (0x36) LPUART1 RX Idle is selected. 110111 - (0x37) LPI2C0 Master Stop is selected. 111000 - (0x38) LPI2C0 Slave STOP is selected. 111001 - (0x39) LPI2C1 Master STOP is selected. 111010 - (0x3A) LPI2C1 Slave STOP is selected. 111011 - (0x3B) LPSPI0 Frame is selected. 111100 - (0x3C) LPSPI0 RX Data is selected. 111101 - (0x3D) LPSPI1 Frame is selected. 111110 - (0x3E) LPSPI1 RX Data is selected. 111111 - (0x3F) SIM Software Trigger is selected.

23.3 Memory map and register definition

The TRGMUX module contains register fields for selecting the trigger input for peripheral modules.

23.3.1 TRGMUX Register Descriptions

These register may not be applicable to all instances of TRGMUX. For more details on the registers supported on each module instance, please refer to "The TRGMUX as implemented on the chip."

Table 23-1. TRGMUX Memory Map

Offset	Register	Width (In bits)	Access	Reset value
40063000h	TRGMUX DMAMUX_CH0 (TRGMUX_DMAMUX_CH0)	32	RW	00000000h
40063004h	TRGMUX XB_OUT0_3 (TRGMUX_XB_OUT0_3)	32	RW	00000000h
40063008h	TRGMUX XB_OUT4_7 (TRGMUX_XB_OUT4_7)	32	RW	00000000h
4006300Ch	TRGMUX ADC0 (TRGMUX_ADC0)	32	RW	00000000h
40063010h	TRGMUX ADC1 (TRGMUX_ADC1)	32	RW	00000000h
4006301Ch	TRGMUX CMP0 (TRGMUX_CMP0)	32	RW	00000000h
40063028h	TRGMUX FTM0 (TRGMUX_FTM0)	32	RW	00000000h
4006302Ch	TRGMUX FTM1 (TRGMUX_FTM1)	32	RW	00000000h

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 23-1. TRGMUX Memory Map (continued)

Offset	Register	Width (In bits)	Access	Reset value
40063030h	TRGMUX FTM2 (TRGMUX_FTM2)	32	RW	00000000h
40063034h	TRGMUX FTM3 (TRGMUX_FTM3)	32	RW	00000000h
40063038h	TRGMUX PDB0 (TRGMUX_PDB0)	32	RW	00000000h
4006303Ch	TRGMUX PDB1 (TRGMUX_PDB1)	32	RW	00000000h
40063044h	TRGMUX FLEXIO (TRGMUX_FLEXIO)	32	RW	00000000h
40063048h	TRGMUX LPIT (TRGMUX_LPIT)	32	RW	00000000h
4006304Ch	TRGMUX LPUART0 (TRGMUX_LPUART0)	32	RW	00000000h
40063050h	TRGMUX LPUART1 (TRGMUX_LPUART1)	32	RW	00000000h
40063054h	TRGMUX LPI2C0 (TRGMUX_LPI2C0)	32	RW	00000000h
40063058h	TRGMUX LPI2C1 (TRGMUX_LPI2C1)	32	RW	00000000h
4006305Ch	TRGMUX LPSPI0 (TRGMUX_LPSPI0)	32	RW	00000000h
40063060h	TRGMUX LPSPI1 (TRGMUX_LPSPI1)	32	RW	00000000h
40063064h	TRGMUX LPTMR0 (TRGMUX_LPTMR0)	32	RW	00000000h

23.3.1.1 TRGMUX Register Descriptions

23.3.1.2 TRGMUX DMAMUX_CH0 (TRGMUX_DMAMUX_CH0)

23.3.1.2.1 Address

Register	Offset
TRGMUX_DMAMUX_CH0	40063000h

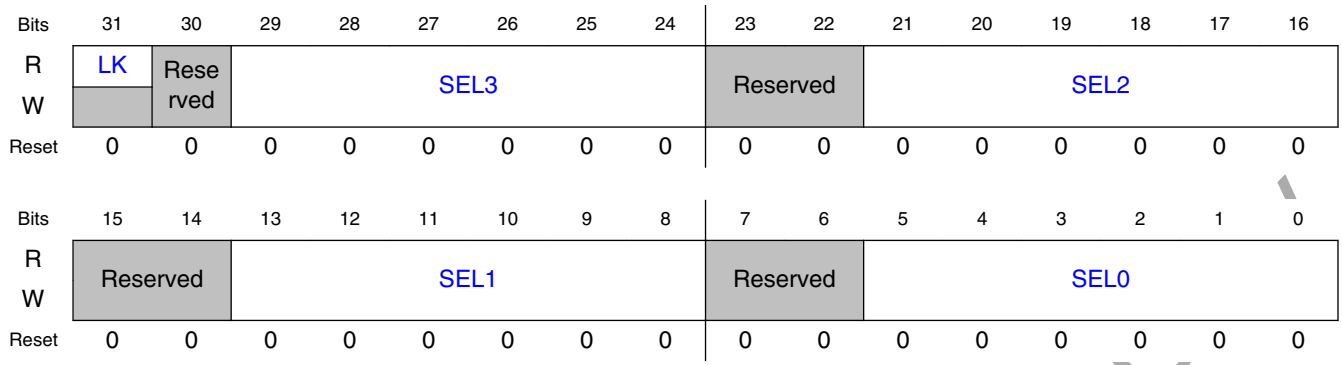
23.3.1.2.2 Function

TRGMUX Register

For Assessment Purp

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.2.3 Diagram



23.3.1.2.4 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.3 TRGMUX XB_OUT0_3 (TRGMUX_XB_OUT0_3)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.3.1 Address

Register	Offset
TRGMUX_XB_OUT0_3	40063004h

TRGMUX Register

23.3.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	SEL3						Reserved		SEL2					
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		SEL1						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.3.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.4 TRGMUX XB_OUT4_7 (TRGMUX_XB_OUT4_7)

23.3.1.4.1 Address

Register	Offset
TRGMUX_XB_OUT4_7	40063008h

TRGMUX Register

23.3.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	SEL3						Reserved		SEL2					
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		SEL1						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.4.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
—	
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

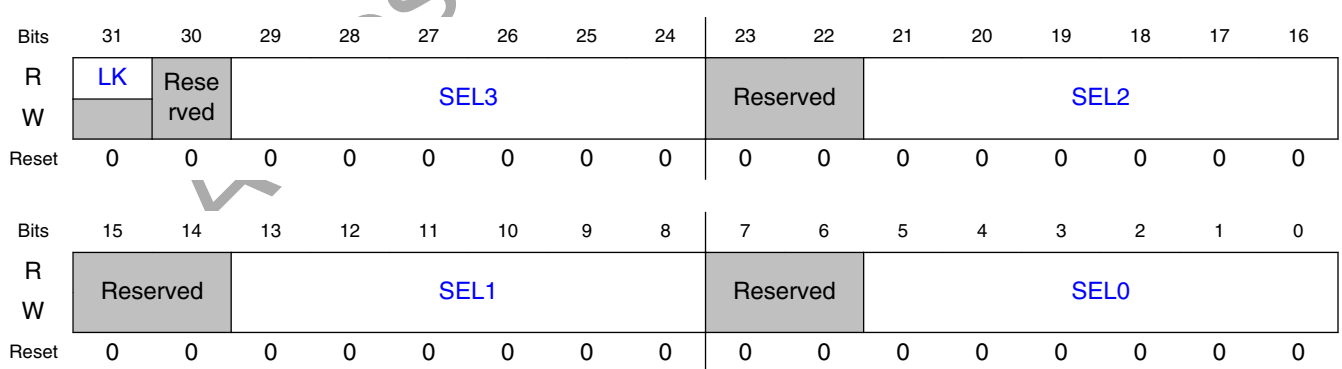
23.3.1.5 TRGMUX ADC0 (TRGMUX_ADC0)

23.3.1.5.1 Address

Register	Offset
TRGMUX_ADC0	4006300Ch

TRGMUX Register

23.3.1.5.2 Diagram



Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.5.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.6 TRGMUX ADC1 (TRGMUX_ADC1)

23.3.1.6.1 Address

Register	Offset
TRGMUX_ADC1	40063010h

TRGMUX Register

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	SEL3						Reserved		SEL2					
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		SEL1						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.6.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.7 TRGMUX CMP0 (TRGMUX_CMP0)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.7.1 Address

Register	Offset
TRGMUX_CMP0	4006301Ch

TRGMUX Register

23.3.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	Reserved						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		Reserved						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.7.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 —	This read-only bit field is reserved and always has the value 0.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 —	This read-only bit field is reserved and always has the value 0.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 —	This read-only bit field is reserved and always has the value 0.
7-6 —	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.8 TRGMUX FTM0 (TRGMUX_FTM0)

23.3.1.8.1 Address

Register	Offset
TRGMUX_FTM0	40063028h

TRGMUX Register

23.3.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	SEL3						Reserved		SEL2					
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		SEL1						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.8.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
—	
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.9 TRGMUX FTM1 (TRGMUX_FTM1)

23.3.1.9.1 Address

Register	Offset
TRGMUX_FTM1	4006302Ch

TRGMUX Register

23.3.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	SEL3						Reserved		SEL2					
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		SEL1						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.9.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.10 TRGMUX FTM2 (TRGMUX_FTM2)

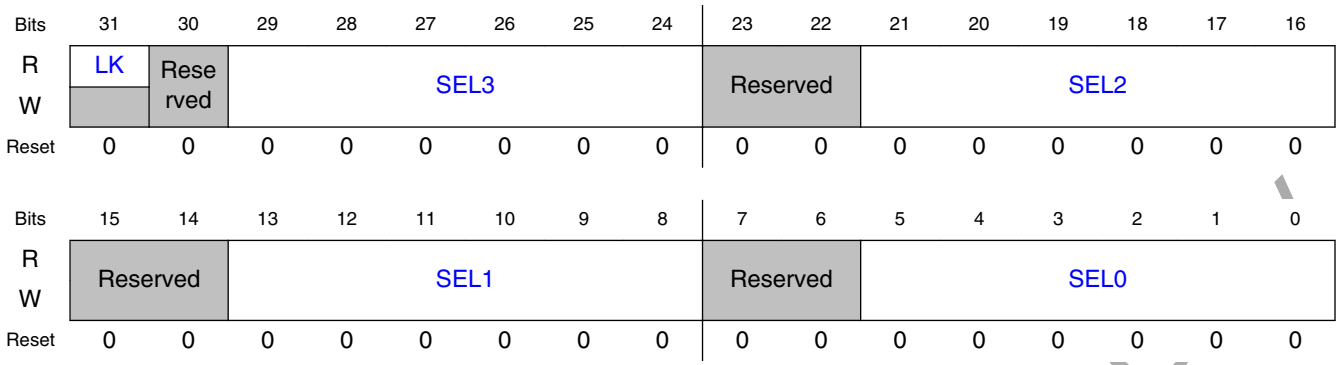
23.3.1.10.1 Address

Register	Offset
TRGMUX_FTM2	40063030h

TRGMUX Register

For Assessment

23.3.1.10.2 Diagram



23.3.1.10.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.11 TRGMUX FTM3 (TRGMUX_FTM3)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.11.1 Address

Register	Offset
TRGMUX_FTM3	40063034h

TRGMUX Register

23.3.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	SEL3						Reserved		SEL2					
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		SEL1						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.11.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.12 TRGMUX PDB0 (TRGMUX_PDB0)

23.3.1.12.1 Address

Register	Offset
TRGMUX_PDB0	40063038h

TRGMUX Register

23.3.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	Reserved						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		Reserved						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.12.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 —	This read-only bit field is reserved and always has the value 0.
23-22	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
—	
21-16 —	This read-only bit field is reserved and always has the value 0.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 —	This read-only bit field is reserved and always has the value 0.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.13 TRGMUX PDB1 (TRGMUX_PDB1)

23.3.1.13.1 Address

Register	Offset
TRGMUX_PDB1	4006303Ch

TRGMUX Register

23.3.1.13.2 Diagram

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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved						Reserved		SEL0					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.13.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 —	This read-only bit field is reserved and always has the value 0.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 —	This read-only bit field is reserved and always has the value 0.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 —	This read-only bit field is reserved and always has the value 0.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.14 TRGMUX FLEXIO (TRGMUX_FLEXIO)

23.3.1.14.1 Address

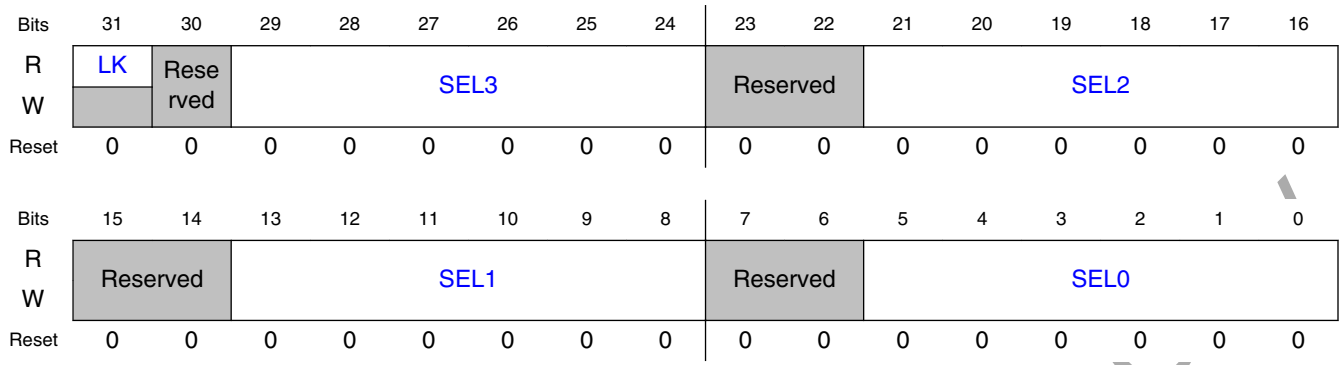
Register	Offset
TRGMUX_FLEXIO	40063044h

TRGMUX Register

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.14.2 Diagram



23.3.1.14.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.15 TRGMUX LPIT (TRGMUX_LPIT)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.15.1 Address

Register	Offset
TRGMUX_LPIT	40063048h

TRGMUX Register

23.3.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	SEL3						Reserved		SEL2					
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		SEL1						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.15.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 SEL3	This read/write bit field is used to configure the MUX select for peripheral trigger input 3. Refer to the Select Bit Fields table in the Features section for bit field information.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 SEL2	This read/write bit field is used to configure the MUX select for peripheral trigger input 2. Refer to the Select Bit Fields table in the Features section for bit field information.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 SEL1	This read/write bit field is used to configure the MUX select for peripheral trigger input 1. Refer to the Select Bit Fields table in the Features section for bit field information.
7-6 —	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.16 TRGMUX LPUART0 (TRGMUX_LPUART0)

23.3.1.16.1 Address

Register	Offset
TRGMUX_LPUART0	4006304Ch

TRGMUX Register

23.3.1.16.2 Diagram

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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved						Reserved		SEL0					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.16.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 —	This read-only bit field is reserved and always has the value 0.
23-22	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
—	
21-16 —	This read-only bit field is reserved and always has the value 0.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 —	This read-only bit field is reserved and always has the value 0.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.17 TRGMUX LPUART1 (TRGMUX_LPUART1)

23.3.1.17.1 Address

Register	Offset
TRGMUX_LPUART1	40063050h

TRGMUX Register

23.3.1.17.2 Diagram

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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved						Reserved		SEL0					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Freescale Confidential Proprietary Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.17.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 —	This read-only bit field is reserved and always has the value 0.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 —	This read-only bit field is reserved and always has the value 0.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 —	This read-only bit field is reserved and always has the value 0.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.18 TRGMUX LPI2C0 (TRGMUX_LPI2C0)

23.3.1.18.1 Address

Register	Offset
TRGMUX_LPI2C0	40063054h

TRGMUX Register

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	Reserved						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		Reserved						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.18.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 —	This read-only bit field is reserved and always has the value 0.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 —	This read-only bit field is reserved and always has the value 0.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 —	This read-only bit field is reserved and always has the value 0.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.19 TRGMUX LPI2C1 (TRGMUX_LPI2C1)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.19.1 Address

Register	Offset
TRGMUX_LPI2C1	40063058h

TRGMUX Register

23.3.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	Reserved						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		Reserved						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.19.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 —	This read-only bit field is reserved and always has the value 0.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 —	This read-only bit field is reserved and always has the value 0.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 —	This read-only bit field is reserved and always has the value 0.
7-6 —	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.20 TRGMUX LPSPI0 (TRGMUX_LPSPI0)

23.3.1.20.1 Address

Register	Offset
TRGMUX_LPSPI0	4006305Ch

TRGMUX Register

23.3.1.20.2 Diagram

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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved						Reserved		SEL0					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.20.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 —	This read-only bit field is reserved and always has the value 0.
23-22	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
—	
21-16 —	This read-only bit field is reserved and always has the value 0.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 —	This read-only bit field is reserved and always has the value 0.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.21 TRGMUX LPSPI1 (TRGMUX_LPSPI1)

23.3.1.21.1 Address

Register	Offset
TRGMUX_LPSPI1	40063060h

TRGMUX Register

23.3.1.21.2 Diagram

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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved						Reserved		SEL0					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Freescale Confidential Proprietary Non-Disclosure Agreement required

23.3.1.21.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 —	This read-only bit field is reserved and always has the value 0.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 —	This read-only bit field is reserved and always has the value 0.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 —	This read-only bit field is reserved and always has the value 0.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

23.3.1.22 TRGMUX LPTMR0 (TRGMUX_LPTMR0)

23.3.1.22.1 Address

Register	Offset
TRGMUX_LPTMR0	40063064h

TRGMUX Register

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

23.3.1.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	Reserved	Reserved						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		Reserved						Reserved		SEL0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.3.1.22.3 Fields

Field	Function
31 LK	This bit shows whether the register can be written or not. 0 - Register can be written. 1 - Register cannot be written until the next system Reset.
30 —	This read-only bit field is reserved and always has the value 0.
29-24 —	This read-only bit field is reserved and always has the value 0.
23-22 —	This read-only bit field is reserved and always has the value 0.
21-16 —	This read-only bit field is reserved and always has the value 0.
15-14 —	This read-only bit field is reserved and always has the value 0.
13-8 —	This read-only bit field is reserved and always has the value 0.
7-6 —	This read-only bit field is reserved and always has the value 0.
5-0 SEL0	This read/write bit field is used to configure the MUX select for peripheral trigger input 0. Refer to the Select Bit Fields table in the Features section for bit field information.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 24
External Watchdog Monitor (EWM)

24.1 Chip-specific External Watchdog Monitor (EWM) information

24.1.1 EWM Memory Map access

Only 8-bit access is supported.

24.1.2 EWM clocks

This table shows the EWM clocks and the corresponding chip clocks.

Table 24-1. EWM clock connections

Module clock	Chip clock
Low Power Clock	128 kHz LPO Clock (LPOCLK)

24.1.3 EWM low-power modes

This table shows the EWM low-power modes and the corresponding chip low-power modes.

Table 24-2. EWM low-power modes

Module mode	Chip mode
Wait	Wait, VLPW
Stop	Stop, VLPS

Assessment Purposes Only

24.1.4 $\overline{\text{EWM\_out}}$ pin state in low power modes

During low power mode the $\overline{\text{EWM\_out}}$ pin preserve its state before entering Wait or Stop mode. When the CPU enters a Run mode from Wait or Stop recovery, the pin resumes its previous state before entering Wait or Stop mode. When the CPU enters Run mode from Power Down, the pin returns to its reset state.

24.2 Introduction

For safety, a redundant watchdog system, External Watchdog Monitor (EWM), is designed to monitor external circuits, as well as the MCU software flow. This provides a back-up mechanism to the internal watchdog that resets the MCU's CPU and peripherals.

The watchdog is generally used to monitor the flow and execution of embedded software within an MCU. The watchdog consists of a counter that if allowed to overflow, forces an internal reset (asynchronous) to all on-chip peripherals and optionally assert the $\overline{\text{RESET}}$ pin to reset external devices/circuits. The overflow of the watchdog counter must not occur if the software code works well and services the watchdog to re-start the actual counter.

The EWM differs from the internal watchdog in that it does not reset the MCU's CPU and peripherals. The EWM provides an independent $\overline{\text{EWM\_out}}$ signal that when asserted resets or places an external circuit into a safe mode. The $\overline{\text{EWM\_out}}$ signal is asserted upon the EWM counter time-out. An optional external input EWM_in is provided to allow additional control of the assertion of $\overline{\text{EWM\_out}}$ signal.

24.2.1 Features

Features of EWM module include:

- Independent LPO_CLK clock source
- Programmable time-out period specified in terms of number of EWM LPO_CLK clock cycles.
- Windowed refresh option
    - Provides robust check that program flow is faster than expected.

For Assessment Purposes Only

  - Programmable window.
  - Refresh outside window leads to assertion of $\overline{\text{EWM\_out}}$.
- Robust refresh mechanism
  - Write values of 0xB4 and 0x2C to EWM Refresh Register within 15 peripheral bus clock cycles.
- One output port, $\overline{\text{EWM\_out}}$, when asserted is used to reset or place the external circuit into safe mode.
- One Input port, EWM_in, allows an external circuit to control the assertion of the $\overline{\text{EWM\_out}}$ signal.

24.2.2 Modes of Operation

This section describes the module's operating modes.

24.2.2.1 Stop Mode

When the EWM is in stop mode, the CPU refreshes to the EWM cannot occur. On entry to stop mode, the EWM's counter freezes.

There are two possible ways to exit from Stop mode:

- On exit from stop mode through a reset, the EWM remains disabled.
- On exit from stop mode by an interrupt, the EWM is re-enabled, and the counter continues to be clocked from the same value prior to entry to stop mode.

Note the following if the EWM enters the stop mode during CPU refresh mechanism: At the exit from stop mode by an interrupt, refresh mechanism state machine starts from the previous state which means, if first refresh command is written correctly and EWM enters the stop mode immediately, the next command has to be written within the next (*EWM_refresh_time*) peripheral bus clocks after exiting from stop mode. User must mask all interrupts prior to executing EWM refresh instructions.

24.2.2.2 Wait Mode

The EWM module treats the stop and wait modes as the same. EWM functionality remains the same in both of these modes.

For Assessment Purposes Only

24.2.2.3 Debug Mode

Entry to debug mode has no effect on the EWM.

- If the EWM is enabled prior to entry of debug mode, it remains enabled.
- If the EWM is disabled prior to entry of debug mode, it remains disabled.

24.2.3 Block Diagram

This figure shows the EWM block diagram.

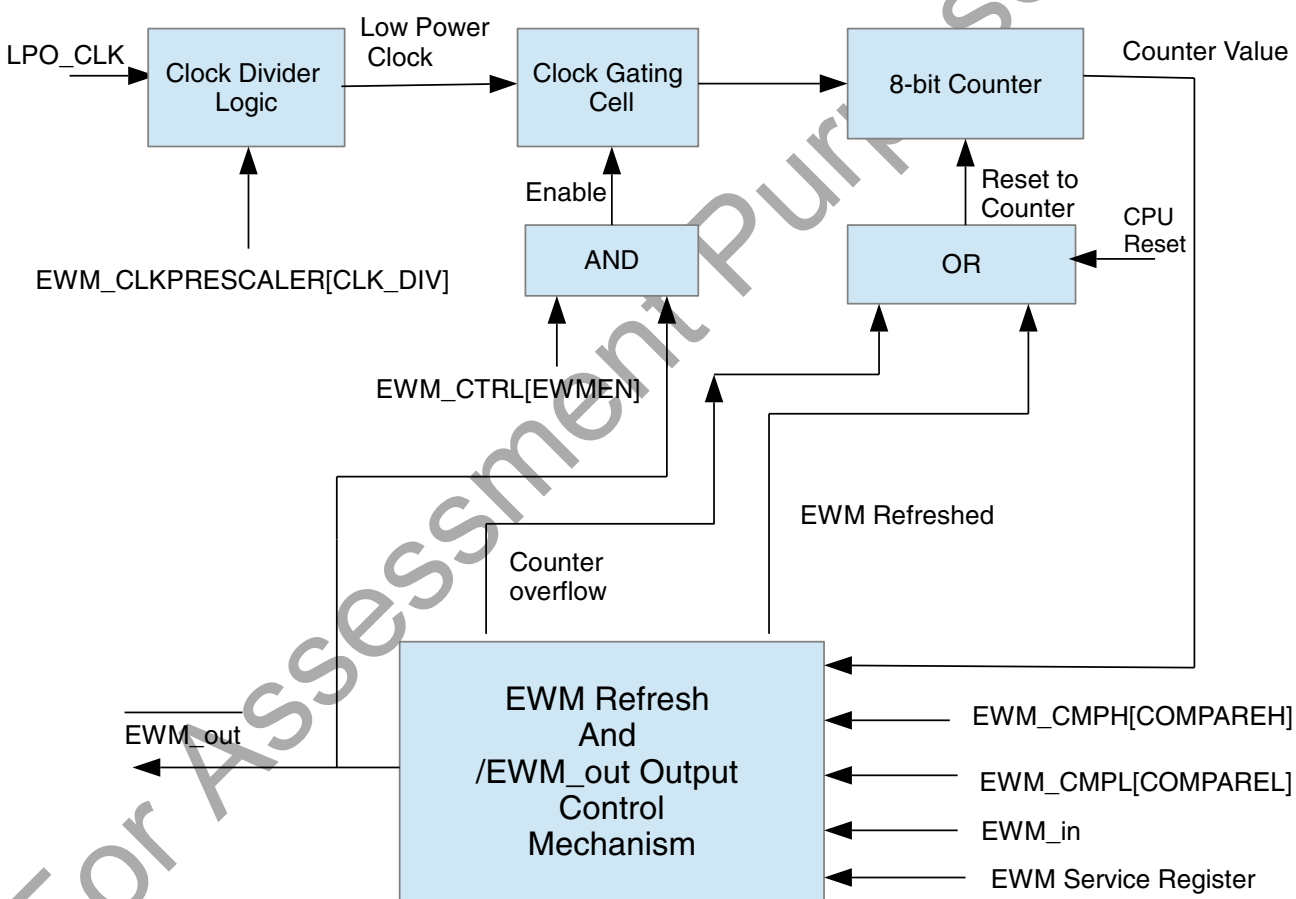


Figure 24-1. EWM Block Diagram

For Assessment Purposes Only

24.3 EWM Signal Descriptions

The EWM has two external signals, as shown in the following table.

Table 24-3. EWM Signal Descriptions

Signal	Description	I/O
EWM_in	EWM input for safety status of external safety circuits. The polarity of EWM_in is programmable using the EWM_CTRL[ASSIN] bit. The default polarity is active-low.	I
$\overline{\text{EWM\_out}}$	EWM reset out signal	O

24.4 Memory Map/Register Definition

This section contains the module memory map and registers.

24.4.1 EWM Register Descriptions

24.4.1.1 EWM Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	Control Register (CTRL)	8	RW	00h
1h	Service Register (SERV)	8	WORZ	00h
2h	Compare Low Register (CMPL)	8	RW	00h
3h	Compare High Register (CMPH)	8	RW	FFh
5h	Clock Prescaler Register (CLKPRESCALER)	8	RW	00h

24.4.1.2 Control Register (CTRL)

For Development Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

24.4.1.2.1 Address

Register	Offset
CTRL	0h

24.4.1.2.2 Function

The CTRL register is cleared by any reset.

NOTE

INEN, ASSIN and EWMEN bits can be written once after a CPU reset. Modifying these bits more than once, generates a bus transfer error.

24.4.1.2.3 Diagram

Bits	0	1	2	3	4	5	6	7
R	EWMEN	ASSIN	INEN	INTEN	0			
W								
Reset	0	0	0	0	0	0	0	0

24.4.1.2.4 Fields

Field	Function
0 EWMEN	EWM enable. This bit when set, enables the EWM module. This resets the EWM counter to zero and deasserts the $\overline{\text{EWM\_out}}$ signal. This bit when unset, keeps the EWM module disabled. It cannot be re-enabled until a next reset, due to the write-once nature of this bit.
1 ASSIN	EWM_in's Assertion State Select. Default assert state of the EWM_in signal is logic zero. Setting the ASSIN bit inverts the assert state of EWM_in signal to a logic one.
2 INEN	Input Enable. This bit when set, enables the EWM_in port.
3 INTEN	Interrupt Enable. This bit when set and $\overline{\text{EWM\_out}}$ is asserted, an interrupt request is generated. To de-assert interrupt request, user should clear this bit by writing 0.
4-7 —	Reserved

nent Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

24.4.1.3 Service Register (SERV)

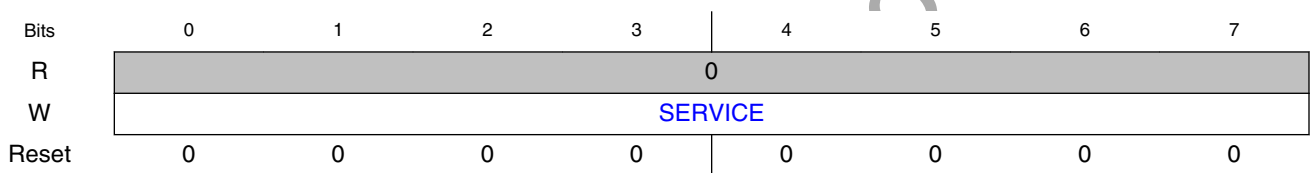
24.4.1.3.1 Address

Register	Offset
SERV	1h

24.4.1.3.2 Function

The SERV register provides the interface from the CPU to the EWM module. It is write-only and reads of this register return zero.

24.4.1.3.3 Diagram



24.4.1.3.4 Fields

Field	Function
0-7 SERVICE	SERVICE The EWM refresh mechanism requires the CPU to write two values to the SERV register: a first data byte of 0xB4, followed by a second data byte of 0x2C. The EWM refresh is invalid if either of the following conditions is true. • The first or second data byte is not written correctly. • The second data byte is not written within a fixed number of peripheral bus cycles of the first data byte. This fixed number of cycles is called *EWM_refresh_time*.

24.4.1.4 Compare Low Register (CMPL)

24.4.1.4.1 Address

Register	Offset
CMPL	2h

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

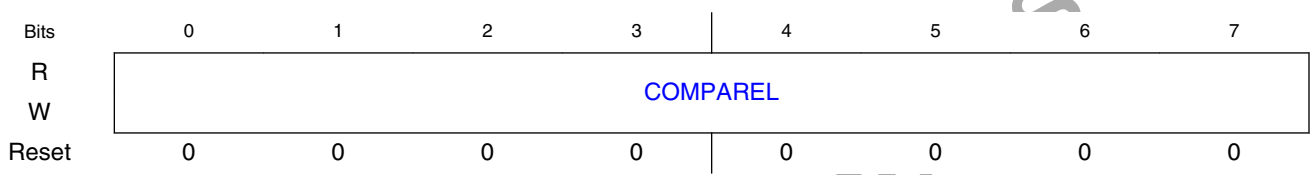
24.4.1.4.2 Function

The CMPL register is reset to zero after a CPU reset. This provides no minimum time for the CPU to refresh the EWM counter.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

24.4.1.4.3 Diagram



24.4.1.4.4 Fields

Field	Function
0-7 COMPAREL	COMPAREL To prevent runaway code from changing this field, software should write to this field after a CPU reset even if the (default) minimum refresh time is required.

24.4.1.5 Compare High Register (CMPH)

24.4.1.5.1 Address

Register	Offset
CMPH	3h

24.4.1.5.2 Function

The CMPH register is reset to 0xFF after a CPU reset. This provides a maximum of 256 clocks time, for the CPU to refresh the EWM counter.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

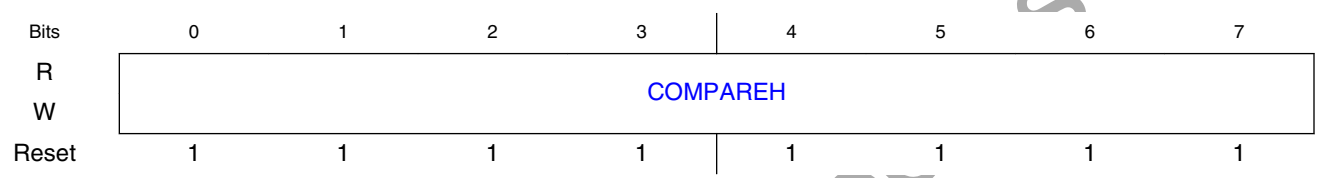
NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

NOTE

The valid values for CMPH are up to 0xFE because the EWM counter never expires when CMPH = 0xFF. The expiration happens only if EWM counter is greater than CMPH.

24.4.1.5.3 Diagram



24.4.1.5.4 Fields

Field	Function
0-7 COMPAREH	COMPAREH To prevent runaway code from changing this field, software should write to this field after a CPU reset even if the (default) maximum refresh time is required.

24.4.1.6 Clock Prescaler Register (CLKPRESCALER)

24.4.1.6.1 Address

Register	Offset
CLKPRESCALER	5h

24.4.1.6.2 Function

This CLKPRESCALER register is reset to 0x00 after a CPU reset.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

NOTE

Write the required prescaler value before enabling the EWM.

NOTE

The implementation of this register is chip-specific. See the Chip Configuration details.

24.4.1.6.3 Diagram

Bits	0	1	2	3	4	5	6	7
R W	CLK_DIV							
Reset	0	0	0	0	0	0	0	0

24.4.1.6.4 Fields

Field	Function
0-7 CLK_DIV	CLK_DIV Selected low power clock source for running the EWM counter can be prescaled as below. • Prescaled clock frequency = low power clock source frequency / (1 + CLK_DIV)

24.5 Functional Description

The following sections describe functional details of the EWM module.

NOTE

When the BUS_CLK is lost, then EWM module doesn't generate the $\overline{\text{EWM\_out}}$ signal and no refresh operation is possible

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

24.5.1 The $\overline{\text{EWM\_out}}$ Signal

The $\overline{\text{EWM\_out}}$ is a digital output signal used to gate an external circuit (application specific) that controls critical safety functions. For example, the $\overline{\text{EWM\_out}}$ could be connected to the high voltage transistors circuits that control an AC motor in a large appliance.

The $\overline{\text{EWM\_out}}$ signal remains deasserted when the EWM is being regularly refreshed by the CPU within the programmable refresh window, indicating that the application code is executed as expected.

The $\overline{\text{EWM\_out}}$ signal is asserted in any of the following conditions:

- The EWM refresh occurs when the counter value is less than CMPL value.
- The EWM counter value reaches the CMPH value, and no EWM refresh has occurred.
- If functionality of EWM_in pin is enabled and EWM_in pin is asserted while refreshing the EWM.
- After any reset (by the virtue of the external pull-down mechanism on the $\overline{\text{EWM\_out}}$ pin)

The $\overline{\text{EWM\_out}}$ is asserted after any reset by the virtue of the external pull-down mechanism on the $\overline{\text{EWM\_out}}$ signal. Then, to deassert the $\overline{\text{EWM\_out}}$ signal, set EWMEN bit in the CTRL register to enable the EWM.

If the $\overline{\text{EWM\_out}}$ signal shares its pad with a digital I/O pin, on reset this actual pad defers to being an input signal. The pad state is controlled by the $\overline{\text{EWM\_out}}$ signal only after the EWM is enabled by the EWMEN bit in the CTRL register.

Note

$\overline{\text{EWM\_out}}$ pad must be in pull down state when EWM functionality is used and when EWM is under Reset.

24.5.2 The EWM_in Signal

The EWM_in is a digital input signal for safety status of external safety circuits, that allows an external circuit to control the assertion of the $\overline{\text{EWM\_out}}$ signal. For example, in the application, an external circuit monitors a critical safety function, and if there is fault with safety function, the external circuit can then actively initiate the $\overline{\text{EWM\_out}}$ signal that controls the gating circuit.

For Assessment Purposes Only

The EWM_in signal is ignored if the EWM is disabled, or if INEN bit of CTRL register is cleared, as after any reset.

On enabling the EWM (setting the CTRL[EWMEN] bit) and enabling EWM_in functionality (setting the CTRL[INEN] bit), the EWM_in signal must be in the deasserted state prior to the CPU start refreshing the EWM. This ensures that the $\overline{\text{EWM\_out}}$ stays in the deasserted state; otherwise, the $\overline{\text{EWM\_out}}$ output signal is asserted.

Note

The user must update the CMPH and CMPL registers prior to enabling the EWM. After enabling the EWM, the counter resets to zero, therefore the user shall provide a reasonable time after a power-on reset for the external monitoring circuit to stabilize. The user shall also ensure that the EWM_in pin is deasserted.

24.5.3 EWM Counter

It is an 8-bit ripple counter fed from a clock source that is independent of the peripheral bus clock source. As the preferred time-out is between 1 ms and 100 ms the actual clock source should be in the kHz range.

The counter is reset to zero after the CPU reset, or when EWM refresh action completes, or at counter overflow. The counter value is not accessible to the CPU.

24.5.4 EWM Compare Registers

The compare registers CMPL and CMPH are write-once after a CPU reset and cannot be modified until another CPU reset occurs.

The EWM compare registers are used to create a refresh window to refresh the EWM module.

It is illegal to program CMPL and CMPH with same value. In this case, as soon as counter reaches (CMPL + 1), $\overline{\text{EWM\_out}}$ is asserted.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

24.5.5 EWM Refresh Mechanism

Other than the initial configuration of the EWM, the CPU can only access the EWM by the EWM Service Register. The CPU must access the EWM service register with correct write of unique data within the windowed time frame as determined by the CMPL and CMPH registers for correct EWM refresh operation. Therefore, three possible conditions can occur:

Table 24-4. EWM Refresh Mechanisms

Condition	Mechanism
An EWM refresh action completes when: CMPL < Counter < CMPH.	The software behaves as expected and the EWM counter is reset to zero. The $\overline{\text{EWM\_out}}$ output signal remains in the deasserted state if, during the EWM refresh action, the EWM_in input has been in deasserted state..
An EWM refresh action completes when Counter < CMPL	The software refreshes the EWM before the windowed time frame, the counter is reset to zero and the $\overline{\text{EWM\_out}}$ output signal is asserted irrespective of the input $\overline{\text{EWM\_in}}$.
Counter value reaches CMPH prior to completion of EWM refresh action.	Software has not refreshed the EWM. The EWM counter is reset to zero and the $\overline{\text{EWM\_out}}$ output signal is asserted irrespective of the input $\overline{\text{EWM\_in}}$.

24.5.6 EWM Interrupt

When $\overline{\text{EWM\_out}}$ is asserted, an interrupt request is generated to indicate the assertion of the EWM reset out signal. This interrupt is enabled when CTRL[INTEN] is set. Clearing this bit clears the interrupt request but does not affect $\overline{\text{EWM\_out}}$. The $\overline{\text{EWM\_out}}$ signal can be deasserted only by forcing a system reset.

24.5.7 Counter clock prescaler

The EWM counter clock source can be prescaled by a clock divider, by programming CLKPRESCALER[CLK_DIV]. This divided clock is used to run the EWM counter.

NOTE

The divided clock used to run the EWM counter must be no more than half the frequency of the bus clock.

For Assessment Purp

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 25
Error Injection Module (EIM)

25.1 Chip-specific EIM information

This chip has one instance of the EIM.

25.1.1 EIM channels

The EIM on this chip supports 2 channels. The number of each channel corresponds to:

- A field in the EIM's EICHEN register
- An EICHD*n*_WORD0 register
- An EICHD*n*_WORD1 register

25.1.2 EIM channel assignments

Each EIM channel corresponds to a specific RAM array target. The EIM channel can inject errors on accesses to that specific RAM array. The following table shows the channel assignments.

Table 25-1. RAM array targets of EIM channels

EIM channel	RAM array target
0	Cortex-M4 TCM lower (bits 31-0)
1	Cortex-M4 TCM upper (bits 31-0)

Preassessment Purposes Only

25.1.3 EIM_EICHD*n*_WORD register bit mapping

For each channel, the following table shows the implemented bits of these registers:

- Error Injection Channel Descriptor, Word0 (EIM_EICHD*n*_WORD0)
- Error Injection Channel Descriptor, Word1 (EIM_EICHD*n*_WORD1)

On this chip, the EIM_EICHD*n*_WORD2 register is not used.

Table 25-2. Checkbit Mask and Data Mask field layout

Off-set	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
100	Cortex-M4 TCM lower CHKBIT_MASK (see Table 25-3)								Reserved																							
104	Cortex-M4 TCM lower B0_3DATA_MASK (bits 31-0)																															
108	Reserved																															
200	Cortex-M4 TCM upper CHKBIT_MASK (see Table 25-3)								Reserved																							
204	Cortex-M4 TCM upper B0_3DATA_MASK (bits 31-0)																															
208	Reserved																															

For CHKBIT_MASK fields implemented for Cortex-M4 TCM RAM arrays, the following table shows the detailed correlation of checkbits to data bits.

Table 25-3. Checkbit Mask field layout details for Cortex-M4 TCM RAM arrays

Offset	31	30	29	28	27	26	25	24
100	Cortex-M4 TCM lower checkbits corresponding to data bits 31-24		Cortex-M4 TCM lower checkbits corresponding to data bits 23-16		Cortex-M4 TCM lower checkbits corresponding to data bits 15-8		Cortex-M4 TCM lower checkbits corresponding to data bits 7-0	
200	Cortex-M4 TCM upper checkbits corresponding to data bits 31-24		Cortex-M4 TCM upper checkbits corresponding to data bits 23-16		Cortex-M4 TCM upper checkbits corresponding to data bits 15-8		Cortex-M4 TCM upper checkbits corresponding to data bits 7-0	

25.2 Introduction

The Error Injection module is mainly used for diagnostic purposes. It provides a method for diagnostic coverage of the peripheral memories. See the chip-specific EIM information to determine which peripheral memories are supported by this method.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

25.2.1 Overview

The Error Injection Module (EIM) provides support for inducing single-bit and multi-bit inversions on read data when accessing peripheral RAMs. Injecting faults on memory accesses can be used to exercise the SEC-DED ECC function of the related system.

NOTE

The following diagram shows an example EIM implementation with a 64-bit read data bus and an 8-bit checkbit bus.

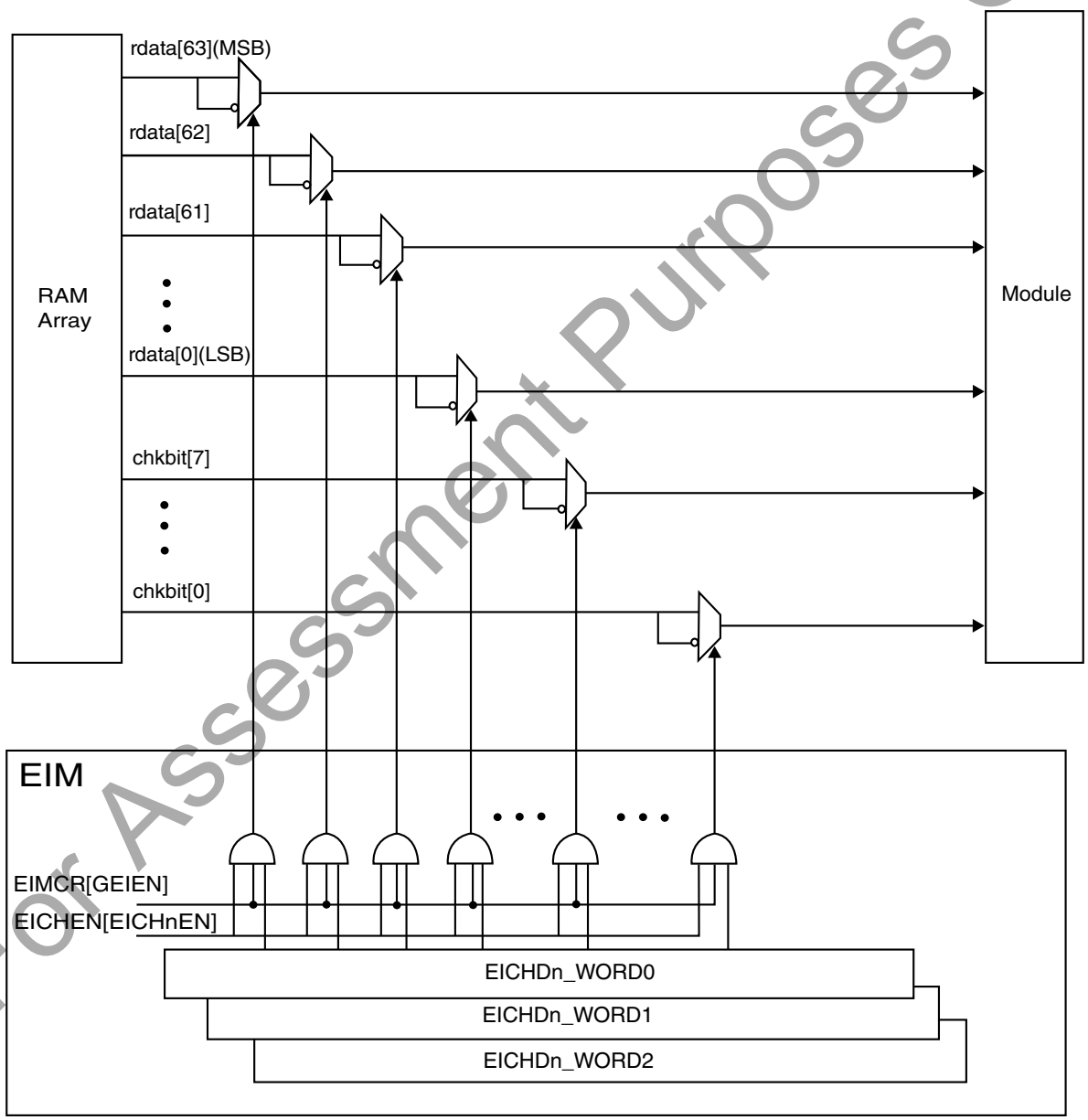


Figure 25-1. EIM functional block diagram (64-bit read data bus and 8-bit checkbit bus)

For Assessment Purposes Only

25.2.2 Features

The EIM includes these features:

- Supports 2 error injection channels
- Protection against accidental enable and reconfiguration error injection function via two-stage enable mechanism

25.3 Memory map and register definition

The EIM provides an IPS programming model mapped to an on-platform peripheral slot.

Programming model access

All system bus masters can access the programming model:

- Only in supervisor mode
- Using only 32-bit (word) accesses

Any of the following attempted references to the programming model generates an IPS error termination:

- In user mode
- Using non-32-bit access sizes
- To undefined (reserved) addresses

Attempted updates to the programming model while the EIM is in the midst of an operation will result in non-deterministic behavior.

Error injection channel descriptor: function and structure

Each error injection channel descriptor:

- Specifies a mask that defines which bits of the read data and checkbit bus from target RAM are inverted on a read access.
- Consists of a 128-bit (16-byte) structure, composed of four 32-bit words, in the EIM programming model.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- Each descriptor consists of Error Injection Channel Descriptor Word0-1 (EICHD*n*_WORD0-1) and, depending on the implementation, Word2 (EICHD*n*_WORD2).
- When Word2 is unused, the corresponding word in the 16-byte structure is unused.
- The fourth word in the 16-byte structure is always unused.

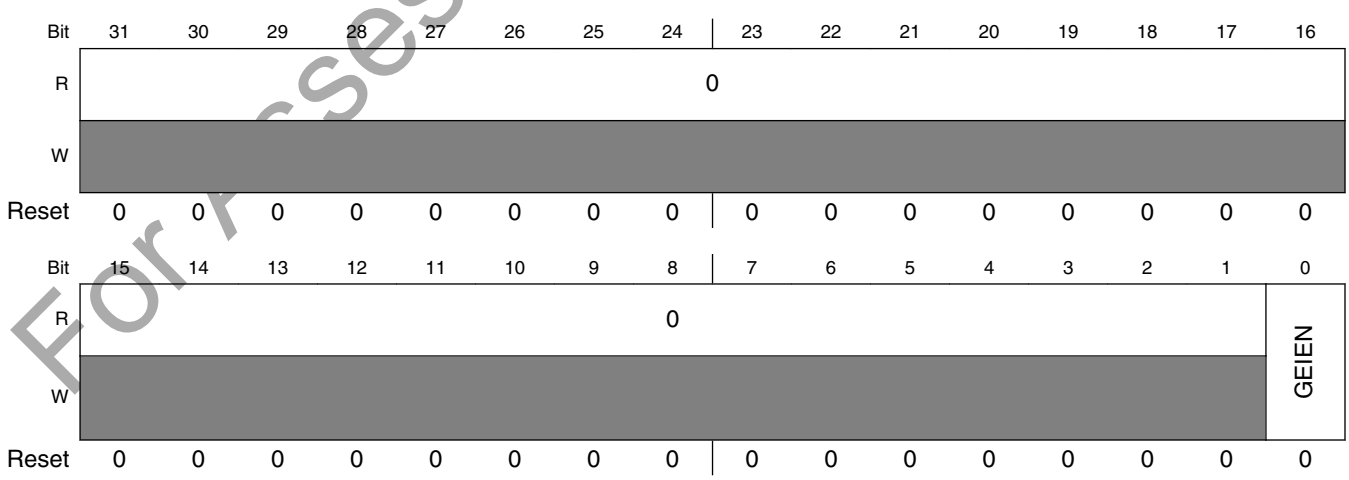
EIM memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Error Injection Module Configuration Register (EIM_EIMCR)	32	R/W	0000_0000h	25.3.1/443
4	Error Injection Channel Enable register (EIM_EICHEN)	32	R/W	0000_0000h	25.3.2/444
100	Error Injection Channel Descriptor, Word0 (EIM_EICHD0_WORD0)	32	R/W	0000_0000h	25.3.3/445
104	Error Injection Channel Descriptor, Word1 (EIM_EICHD0_WORD1)	32	R/W	0000_0000h	25.3.4/446
200	Error Injection Channel Descriptor, Word0 (EIM_EICHD1_WORD0)	32	R/W	0000_0000h	25.3.3/445
204	Error Injection Channel Descriptor, Word1 (EIM_EICHD1_WORD1)	32	R/W	0000_0000h	25.3.4/446

25.3.1 Error Injection Module Configuration Register (EIM_EIMCR)

The EIM Configuration Register is used to globally enable/disable the error injection function.

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



For Assessment Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

EIM_EIMCR field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 GEIEN	Global Error Injection Enable This bit globally enables or disables the error injection function of the EIM. This field is initialized by hardware reset. 0 Disabled 1 Enabled

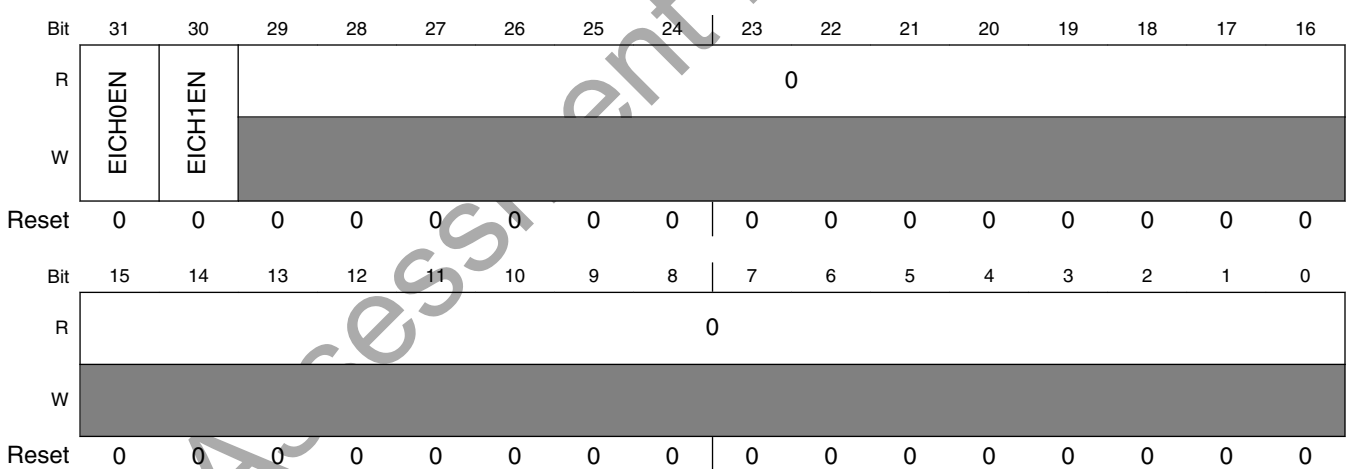
25.3.2 Error Injection Channel Enable register (EIM_EICHEN)

Each field of the Error Injection Channel Enable register (EICHEN) is used to enable or disable the corresponding error injection channel.

NOTE

To enable an error injection channel, the Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted.

Address: 0h base + 4h offset = 4h



EIM_EICHEN field descriptions

Field	Description
31 EICH0EN	Error Injection Channel 0 Enable This field enables the corresponding error injection channel. The Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted to enable error injection. After error injection is enabled, all subsequent read accesses incur one or more bit inversions as defined in the corresponding EICHD*n*_WORD registers. Error injection remains in effect until the error injection channel is manually disabled via software. Any write to the corresponding EICHD*n*_WORD registers clears the corresponding EICHEN[EICH*n*EN] field, disabling the error injection channel.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Assessment Purposes C

EIM_EICHEN field descriptions (continued)

Field	Description
	0 Error injection is disabled on Error Injection Channel 0 1 Error injection is enabled on Error Injection Channel 0
30 EICH1EN	Error Injection Channel 1 Enable This field enables the corresponding error injection channel. The Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted to enable error injection. After error injection is enabled, all subsequent read accesses incur one or more bit inversions as defined in the corresponding EICHD*n*_WORD registers. Error injection remains in effect until the error injection channel is manually disabled via software. Any write to the corresponding EICHD*n*_WORD registers clears the corresponding EICHEN[EICH*n*EN] field, disabling the error injection channel. 0 Error injection is disabled on Error Injection Channel 1 1 Error injection is enabled on Error Injection Channel 1
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

25.3.3 Error Injection Channel Descriptor, Word0 (EIM_EICHD*n*_WORD0)

The first word of the Error Injection Channel Descriptor defines a left-justified mask field (CHKBIT_MASK) whose width is up to 8 bits. Each bit of CHKBIT_MASK specifies whether the corresponding bit of the checkbit bus from the target RAM should be inverted or remain unmodified on read accesses. Successful writes to this field clear the corresponding error injection channel valid bit, EICHEN[EICH*n*EN].

Address: 0h base + 100h offset + (256d × 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	CHKBIT_MASK								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

EIM_EICHD*n*_WORD0 field descriptions

Field	Description
31–24 CHKBIT_MASK	Checkbit Mask This field defines a bit-mapped mask that specifies whether the corresponding bit of the checkbit bus from the target RAM should be inverted or remain unmodified. The width of the field for the channel is defined in the chip-specific EIM information. 0 The corresponding bit of the checkbit bus remains unmodified. 1 The corresponding bit of the checkbit bus is inverted.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Assessment Purpo

EIM_EICHD*n*_WORD0 field descriptions (continued)

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

25.3.4 Error Injection Channel Descriptor, Word1 (EIM_EICHD*n*_WORD1)

The second word of the Error Injection Channel Descriptor defines the bits in the mask corresponding to bytes 0–3 of the read data bus. Each bit specifies whether the corresponding bit of the read data bus from the target RAM should be inverted or remain unmodified on read accesses. Successful writes to this field clear the corresponding error injection channel valid field, EICHEN[EICH*n*EN].

Address: 0h base + 104h offset + (256d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	B0_3DATA_MASK																															
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

EIM_EICHD*n*_WORD1 field descriptions

Field	Description
B0_3DATA_MASK	Data Mask Bytes 0-3 This field defines a bit-mapped mask that specifies whether the corresponding bit of the read data bus from the target RAM should be inverted or remain unmodified. For details about which bits in this field are implemented for the applicable channel and how the bits map to bytes 0-3 of the read data bus, see the chip-specific EIM information. 0 The corresponding bit of bytes 0-3 on the read data bus remains unmodified 1 The corresponding bit of bytes 0-3 on the read data bus is inverted.

25.4 Functional description

The EIM provides protection against accidental enabling and reconfiguration of the error injection function by enforcing a two-stage enablement mechanism. To properly enable the error injection mechanism for a channel, execute both of the following:

- Assert the EICHEN[EICH*n*EN] field, where *n* denotes the channel number.
- Assert EIMCR[GEIEN].

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

NOTE

When the use case for a channel requires writing any EICHD*n*_WORD register, write the EICHD*n*_WORD register before executing the two-stage enablement mechanism. A successful write to any EICHD*n*_WORD register clears the corresponding EICHEN[EICH*n*EN] field.

The EIM supports 2 error injection channels. Each channel:

- is assigned to a single memory array interface.
- intercepts the assigned memory read data bus and checkbit bus and injects errors by inverting the value transmitted for selected bits on each bus line.

On a memory read access, the applicable EICHD*n*_WORD registers define which bits of the read data and/or checkbit bus to invert.

Figure 25-1 depicts the interception and override of a 64-bit read data bus and an 8-bit checkbit data bus for an example memory array.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 26
Error Reporting Module (ERM)

26.1 Chip-specific ERM information

This chip has one instance of the ERM.

26.1.1 ERM memory channels

The ERM on this chip supports 2 memory channels numbered 0-1. The number of each memory channel, *n*, corresponds to:

- An ENCIE*n* field and an ESCIE*n* field in the ERM Configuration Register (ERM_CR)
- An NCE*n* field and an SBC*n* field in the ERM Status Register (ERM_SR)
- An ERM Memory *n* Error Address Register (ERM_EAR*n*)

On this chip, the 32-bit-width spaces at offsets 104h and 114h are read-only and always read as 0.

26.1.2 Sources of memory error events

Each ERM channel *n* corresponds to a source of potential memory error events. The following table shows the channel assignments.

Table 26-1. Memory error event sources for ERM channels

ERM channel *n*	Memory *n* event source[1]
0	Cortex-M4 TCM lower
1	Cortex-M4 TCM upper

1. Throughout this chapter, "Memory n" designates the memory array sourced by ERM channel n.

For Assessment Purposes Only

26.2 Introduction

26.2.1 Overview

The Error Reporting Module (ERM) provides information and optional interrupt notification on memory error events associated with ECC (Error Correction Code). The ERM collects ECC events on memory accesses for platform local memory arrays, such as flash memory, system RAM, or peripheral RAMs. See the chip-specific ERM information for details about supported memory sources and specific memory channel assignments.

26.2.2 Features

The ERM includes these features:

- Capture address information on single-bit correction and non-correctable ECC events
- Optional interrupt notification on captured ECC events
- Support for ECC event capturing for memory sources, with individual reporting fields and interrupt configuration per memory channel.

26.3 Memory map and register definition

The ERM provides an IPS programming model mapped to an on-platform peripheral slot. The programming model can only be referenced using a 32-bit (word) access. Attempted references using different access sizes or to undefined (reserved) addresses or in user mode generates an IPS error termination. The ERM allows access to the programming model by all system bus masters.

The programming model can only be accessed in supervisor mode.

Attempted updates to the programming model while the ERM module is in the middle of an operation will result in non-deterministic behavior.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

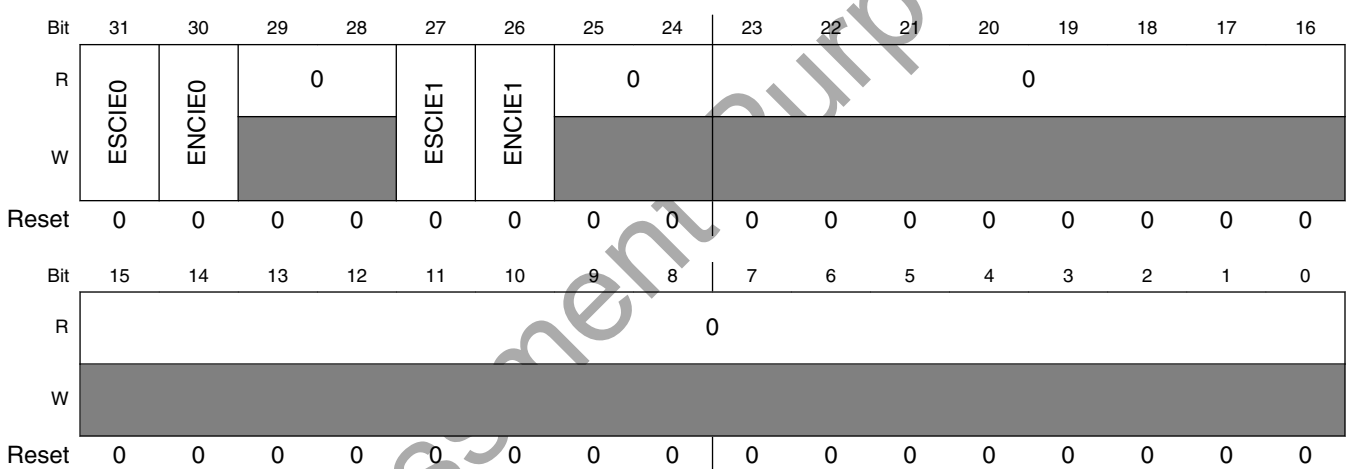
ERM memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	ERM Configuration Register (ERM_CR)	32	R/W	0000_0000h	26.3.1/451
10	ERM Status Register (ERM_SR)	32	w1c	0000_0000h	26.3.2/453
100	ERM Memory n Error Address Register (ERM_EAR0)	32	R	0000_0000h	26.3.3/454
110	ERM Memory n Error Address Register (ERM_EAR1)	32	R	0000_0000h	26.3.3/454

26.3.1 ERM Configuration Register (ERM_CR)

The ERM Configuration Register is a 32-bit control register for configuring the interrupt notification capability for memory channels 0 to 1.

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



ERM_CR field descriptions

Field	Description
31 ESCIE0	Enable Memory 0 Single Correction Interrupt Notification This field is initialized by hardware reset. **NOTE:** Refer to the chip-specific ERM information for details on Memory 0 mapping. 0 Interrupt notification of Memory 0 single-bit correction events is disabled. 1 Interrupt notification of Memory 0 single-bit correction events is enabled.
30 ENCIE0	Enable Memory 0 Non-Correctable Interrupt Notification This field is initialized by hardware reset. **NOTE:** Refer to the chip-specific ERM information for details on Memory 0 mapping. 0 Interrupt notification of Memory 0 non-correctable error events is disabled. 1 Interrupt notification of Memory 0 non-correctable error events is enabled.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ERM_CR field descriptions (continued)

Field	Description
29–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 ESCIE1	Enable Memory 1 Single Correction Interrupt Notification This field is initialized by hardware reset. **NOTE:** Refer to the chip-specific ERM information for details on Memory 1 mapping. 0 Interrupt notification of Memory 1 single-bit correction events is disabled. 1 Interrupt notification of Memory 1 single-bit correction events is enabled.
26 ENCIE1	Enable Memory 1 Non-Correctable Interrupt Notification This field is initialized by hardware reset. **NOTE:** Refer to the chip-specific ERM information for details on Memory 1 mapping. 0 Interrupt notification of Memory 1 non-correctable error events is disabled. 1 Interrupt notification of Memory 1 non-correctable error events is enabled.
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

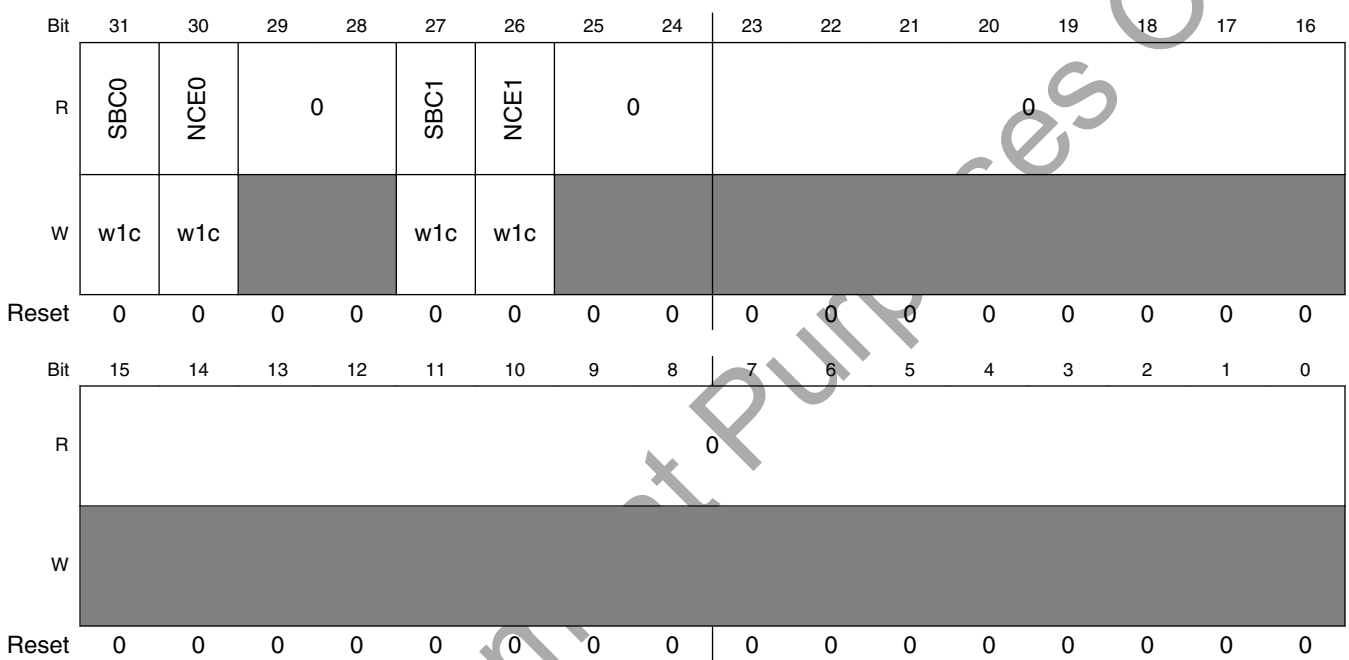
For Assessment Pu

Freescale Confidential Proprietary - Non-Disclosure Agreement required

26.3.2 ERM Status Register (ERM_SR)

The ERM Status Register is a 32-bit control register for signaling which types of ECC events have been detected for each memory channel. The register signals the last memory event to be detected for memory channels 0 to 1.

Address: 0h base + 10h offset = 10h



ERM_SR field descriptions

Field	Description
31 SBC0	Memory 0 Single-Bit Correction Event This field is initialized by hardware reset. Write 1 to clear this field. This write also deasserts the corresponding interrupt notification if ERM_CR[ESCIE0] is enabled. **NOTE:** Refer to the chip-specific ERM information for details on Memory 0 mapping. 0 No single-bit correction event on Memory 0 detected 1 Single-bit correction event on Memory 0 detected
30 NCE0	Memory 0 Non-Correctable Error Event This field is initialized by hardware reset. Write 1 to clear this field. This write also deasserts the corresponding interrupt notification if ERM_CR[ENCIE0] is enabled. **NOTE:** Refer to the chip-specific ERM information for details on Memory 0 mapping.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ERM_SR field descriptions (continued)

Field	Description
	0 No non-correctable error event on Memory 0 detected 1 Non-correctable error event on Memory 0 detected
29–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 SBC1	Memory 1 Single-Bit Correction Event This field is initialized by hardware reset. Write 1 to clear this field. This write also deasserts the corresponding interrupt notification if ERM_CR[ESCIE1] is enabled. **NOTE:** Refer to the chip-specific ERM information for details on Memory 1 mapping. 0 No single-bit correction event on Memory 1 detected 1 Single-bit correction event on Memory 1 detected
26 NCE1	Memory 1 Non-Correctable Error Event This field is initialized by hardware reset. Write 1 to clear this field. This write also deasserts the corresponding interrupt notification if ERM_CR[ENCIE1] is enabled. **NOTE:** Refer to the chip-specific ERM information for details on Memory 1 mapping. 0 No non-correctable error event on Memory 1 detected 1 Non-correctable error event on Memory 1 detected
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

For Assessment

26.3.3 ERM Memory *n* Error Address Register (ERM_EAR*n*)

The ERM_EAR*n* is a 32-bit register for capturing the address of the last ECC event in Memory *n*, where *n* denotes the memory channel. Any attempted write to ERM_EAR*n* is ignored.

Address: 0h base + 100h offset + (16d × i), where i=0d to 1d

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ERM_EAR*n* field descriptions

Field	Description
EAR	Memory *n* Error Address - This field contains the faulting system address of the last recorded ECC event on Memory *n*. **NOTE:** Refer to the chip-specific ERM information for details on Memory *n* mapping.

26.4 Functional description

26.4.1 Single-bit correction events

Single-bit correction events reported from Memory *n* are recorded in the Error Reporting Module Status Register (ERM_SR[SBC*n*]). The corresponding access address which initiated the ECC event is stored in the Memory *n* Error Address Register (ERM_EAR*n*). The ERM can be optionally configured to generate an interrupt notification upon reporting of a single-bit ECC correction on Memory *n* by enabling ERM_CR[ESCIE*n*]. Writing 1 to ERM_SR[SBC*n*] clears this field and deasserts the interrupt notification if interrupt notification is enabled for this memory channel.

The ERM holds event information only for the last reported event.

26.4.2 Non-correctable error events

Non-correctable ECC error events reported from Memory *n* are recorded in the Error Reporting Module Status Register (ERM_SR[SBC*n*]). The corresponding access address which initiated the ECC event is stored in the Memory *n* Error Address Register (ERM_EAR*n*).

The ERM only holds event information for the last reported event.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 27
Watchdog timer (WDOG)

27.1 Chip-specific Watchdog timer (WDOG) information

27.1.1 WDOG clocks

The WDOG module has following selectable clock sources:

- Internal low power oscillator (LPOCLK)
- Internal Slow IRC clock (SIRC)
- System oscillator clock (SOSC)
- Bus clock

NOTE

WDOG_CNT reset read value can vary depending on timestamp since it is a default running counter.

27.1.2 WDOG low-power modes

This table shows the WDOG low-power modes and the corresponding chip low-power modes.

Table 27-1. WDOG low-power modes

Module mode	Chip mode
Wait	Wait, VLPW
Stop	Stop, VLPS

For Assessment Purposes Only

27.2 Introduction

The Watchdog Timer (WDOG) module is an independent timer that is available for system use. It provides a safety feature to ensure that software is executing as planned and that the CPU is not stuck in an infinite loop or executing unintended code. If the WDOG module is not serviced (refreshed) within a certain period, it resets the MCU.

27.2.1 Features

Features of the WDOG module include:

- Configurable clock source inputs independent from the bus clock
  - Bus clock (slow clock)
  - LPO clock (from PMC)
  - SIRC (8 MHz IRC from SCG)
  - ERCLK (external reference clock from SCG)
- Programmable timeout period
  - Programmable 16-bit timeout value
  - Optional fixed 256 clock prescaler when longer timeout periods are needed
- Robust write sequence for counter refresh
  - Refresh sequence of writing 0xA602 and then 0xB480 within 16 bus clocks
- Window mode option for the refresh mechanism
  - Programmable 16-bit window value
  - Provides robust check that program flow is faster than expected
  - Early refresh attempts trigger a reset.
- Optional timeout interrupt to allow post-processing diagnostics
  - Interrupt request to CPU with interrupt vector for an interrupt service routine (ISR)
  - Forced reset occurs 128 bus clocks after the interrupt vector fetch.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- Configuration bits are write-once-after-reset to ensure watchdog configuration cannot be mistakenly altered.
- Robust write sequence for unlocking write-once configuration bits
  - Unlock sequence of writing 0xC520 and then 0xD928 within 16 bus clocks for allowing updates to write-once configuration bits
  - Software must make updates within 128 bus clocks after unlocking and before WDOG closing unlock window.

27.2.2 Block diagram

The following figure shows a block diagram of the WDOG module.

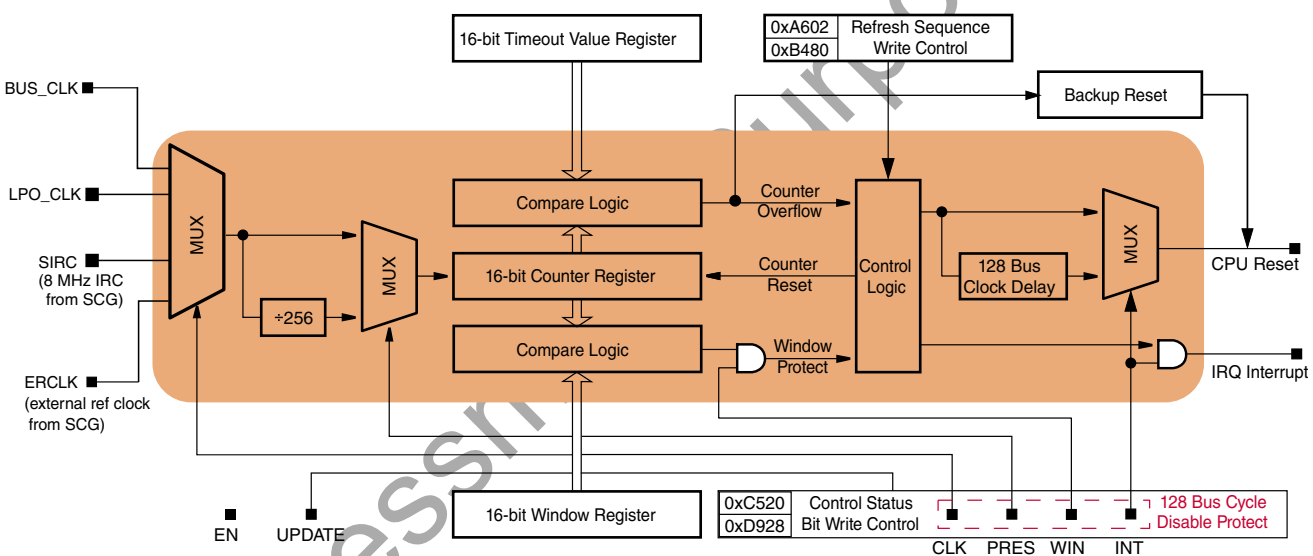


Figure 27-1. WDOG block diagram

27.3 Memory map and register definition

WDOG memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Watchdog Control and Status Register (WDOG_CS)	32	R/W	See section	27.3.1/460
4	Watchdog Counter Register (WDOG_CNT)	32	R/W	0000_0000h	27.3.2/462
8	Watchdog Timeout Value Register (WDOG_TOVAL)	32	R/W	0000_0400h	27.3.3/463
C	Watchdog Window Register (WDOG_WIN)	32	R/W	0000_0000h	27.3.4/464

For Assessment Purposes Only

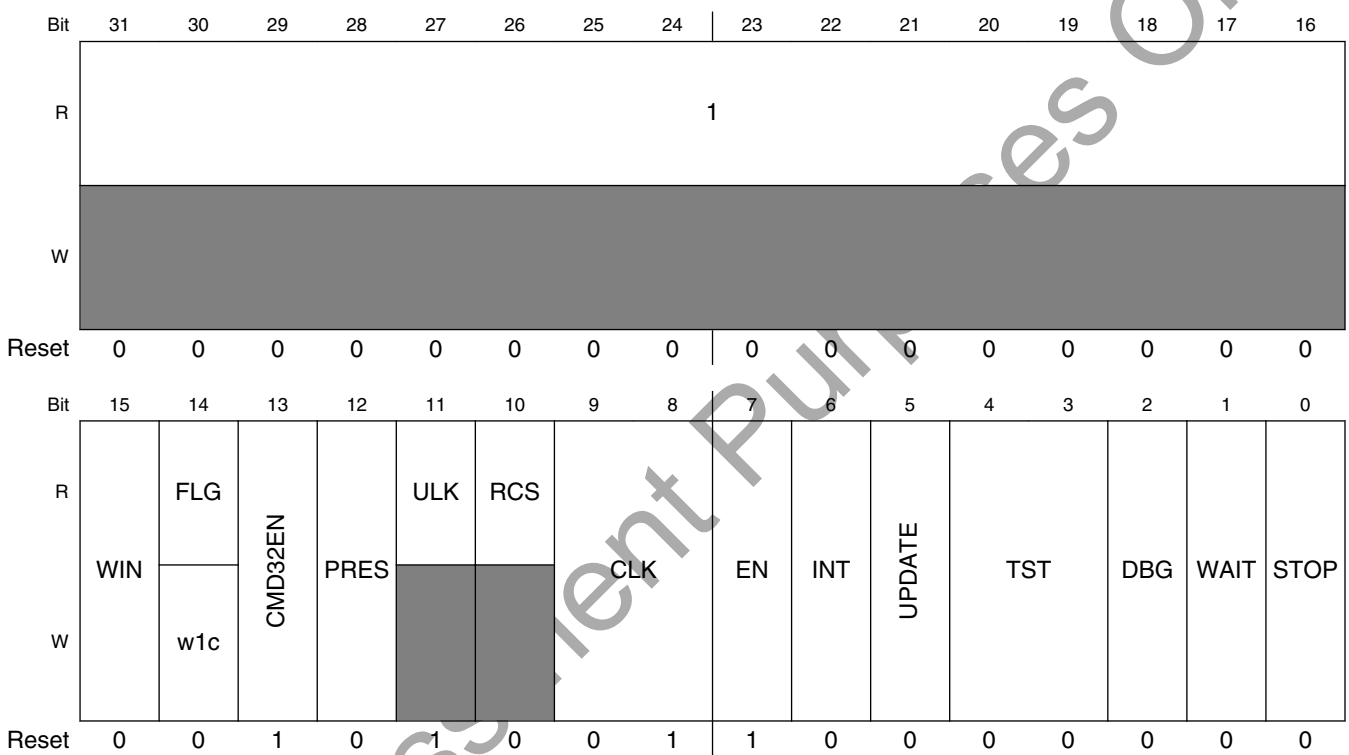
27.3.1 Watchdog Control and Status Register (WDOG_CS)

This section describes the function of Watchdog Control and Status Register.

NOTE

TST is cleared (0:0) on POR only. Any other reset does not affect the value of this field.

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



Freescale Confidential Proprietary Only

WDOG_CS field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
15 WIN	Watchdog Window This write-once bit enables window mode. See the Window mode section. 0 Window mode disabled. 1 Window mode enabled.
14 FLG	Watchdog Interrupt Flag This bit is an interrupt indicator when INT is set in control and status register 1. Write 1 to clear it. 0 No interrupt occurred. 1 An interrupt occurred.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

WDOG_CS field descriptions (continued)

Field	Description
13 CMD32EN	Enables or disables WDOG support for 32-bit (or 16-bit or 8-bit) refresh/unlock command write words 0 Disables support for 32-bit (or 16-bit or 8-bit) refresh/unlock command write words 1 Enables support for 32-bit (or 16-bit or 8-bit) refresh/unlock command write words
12 PRES	Watchdog prescaler This write-once bit enables a fixed 256 pre-scaling of watchdog counter reference clock. (The block diagram shows this clock divider option.) 0 256 prescaler disabled. 1 256 prescaler enabled.
11 ULK	Unlock status This read-only bit indicates whether WDOG is unlocked or not. Default reset value is 1. 0 WDOG is locked. 1 WDOG is unlocked.
10 RCS	Reconfiguration Success This read-only bit indicates whether the reconfiguration is successful or not. Default reset value is 0.This bit is set when new configuration takes effect, and is cleared by successful unlock command. 0 Reconfiguring WDOG. 1 Reconfiguration is successful.
9–8 CLK	Watchdog Clock This write-once field indicates the clock source that feeds the watchdog counter. See the Clock source section. 00 Bus clock 01 LPO clock (128k LPO, 32k LPO, 1k LPO, or 32k OSC), whose selection is controlled via SIM_LPOCLKS[LPOCLKSEL]. 10 System oscillator clock (SOSC, from SCG) 11 Slow internal reference clock (SIRC, from SCG)
7 EN	Watchdog Enable This write-once bit enables the watchdog counter to start counting. 0 Watchdog disabled. 1 Watchdog enabled.
6 INT	Watchdog Interrupt This write-once bit configures the watchdog to immediately generate an interrupt request upon a reset-triggering event (timeout or illegal write to the watchdog), before forcing a reset. After the interrupt vector fetch (which comes after the reset-triggering event), the reset occurs after a delay of 128 bus clocks. 0 Watchdog interrupts are disabled. Watchdog resets are not delayed. 1 Watchdog interrupts are enabled. Watchdog resets are delayed by 128 bus clocks from the interrupt vector fetch.
5 UPDATE	Allow updates This write-once bit allows software to reconfigure the watchdog without a reset.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

WDOG_CS field descriptions (continued)

Field	Description
	0 Updates not allowed. After the initial configuration, the watchdog cannot be later modified without forcing a reset. 1 Updates allowed. Software can modify the watchdog configuration registers within 128 bus clocks after performing the unlock write sequence.
4–3 TST	Watchdog Test Enables the fast test mode. The test mode allows software to exercise all bits of the counter to demonstrate that the watchdog is functioning properly. See the Fast testing of the watchdog section. This write-once field is cleared (0:0) on POR only. Any other reset does not affect the value of this field. 00 Watchdog test mode disabled. 01 Watchdog user mode enabled. (Watchdog test mode disabled.) After testing the watchdog, software should use this setting to indicate that the watchdog is functioning normally in user mode. 10 Watchdog test mode enabled, only the low byte is used. CNT[CNTLOW] is compared with TOVAL[TOVALLOW]. 11 Watchdog test mode enabled, only the high byte is used. CNT[CNTHIGH] is compared with TOVAL[TOVALHIGH].
2 DBG	Debug Enable This write-once bit enables the watchdog to operate when the chip is in debug mode. 0 Watchdog disabled in chip debug mode. 1 Watchdog enabled in chip debug mode.
1 WAIT	Wait Enable This write-once bit enables the watchdog to operate when the chip is in wait mode. 0 Watchdog disabled in chip wait mode. 1 Watchdog enabled in chip wait mode.
0 STOP	Stop Enable This write-once bit enables the watchdog to operate when the chip is in stop mode. 0 Watchdog disabled in chip stop mode. 1 Watchdog enabled in chip stop mode.

For Ass

27.3.2 Watchdog Counter Register (WDOG_CNT)

This section describes the watchdog counter register.

The watchdog counter register provides access to the value of the free-running watchdog counter. Software can read the counter register at any time.

Software cannot write directly to the watchdog counter; however, two write sequences to these registers have special functions:

1. The *refresh sequence* resets the watchdog counter to 0x0000. See the Refreshing the Watchdog section.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

2. The *unlock sequence* allows the watchdog to be reconfigured without forcing a reset (when CS[UPDATE] = 1). See the Example code: Reconfiguring the Watchdog section.

NOTE

All other writes to this register are illegal and force a reset.

Address: 0h base + 4h offset = 4h

Bit	31	30	29	28	27	26	25	24	23	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																CNTHIGH								CNTLOW							
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

WDOG_CNT 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 CNTHIGH	High byte of the Watchdog Counter
CNTLOW	Low byte of the Watchdog Counter

27.3.3 Watchdog Timeout Value Register (WDOG_TOVAL)

This section describes the watchdog timeout value register. TOVAL contains the 16-bit value used to set the timeout period of the watchdog.

The watchdog counter (CNT) is continuously compared with the timeout value (TOVAL). If the counter reaches the timeout value, the watchdog forces a reset.

NOTE

Do not write 0 to the Watchdog Timeout Value Register; otherwise, the watchdog always generates a reset.

Address: 0h base + 8h offset = 8h

Bit	31	30	29	28	27	26	25	24	23	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																TOVALHIGH								TOVALLOW							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

WDOG_TOVAL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

WDOG_TOVAL field descriptions (continued)

Field	Description
15–8 TOVALHIGH	High byte of the timeout value
TOVALLOW	Low byte of the timeout value

27.3.4 Watchdog Window Register (WDOG_WIN)

This section describes the watchdog window register. When window mode is enabled (CS[WIN] is set), The WIN register determines the earliest time that a refresh sequence is considered valid. See the Watchdog refresh mechanism section.

The WIN register value must be less than the TOVAL register value.

Address: 0h base + Ch offset = Ch

Bit	31	30	29	28	27	26	25	24	23	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																WINHIGH								WINLOW							
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

WDOG_WIN 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 WINHIGH	High byte of Watchdog Window
WINLOW	Low byte of Watchdog Window

27.4 Functional description

The WDOG module provides a fail safe mechanism to ensure the system can be reset to a known state of operation in case of system failure, such as the CPU clock stopping or there being a run away condition in the software code. The watchdog counter runs continuously off a selectable clock source and expects to be serviced (refreshed) periodically. If it is not, it resets the system.

The timeout period, window mode, and clock source are all programmable but must be configured within 128 bus clocks after a reset.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

27.4.1 Watchdog refresh mechanism

The watchdog resets the MCU if the watchdog counter is not refreshed. A robust refresh mechanism makes it very unlikely that the watchdog can be refreshed by runaway code.

To refresh the watchdog counter, software must execute a refresh write sequence before the timeout period expires. In addition, if window mode is used, software must not start the refresh sequence until after the time value set in the WIN register. See the following figure.

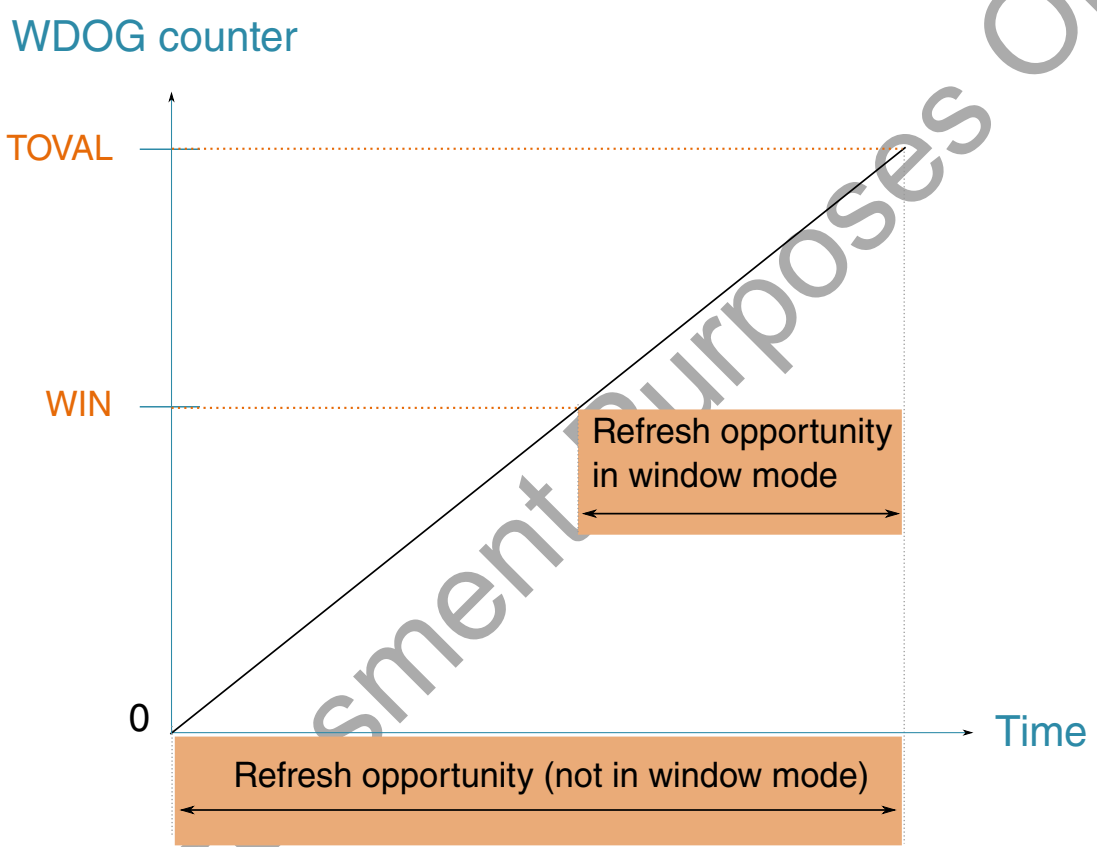


Figure 27-2. Refresh opportunity for the Watchdog counter

27.4.1.1 Window mode

Software finishing its main control loop faster than expected could be an indication of a problem. Depending on the requirements of the application, the WDOG can be programmed to force a reset when refresh attempts are early.

For Assessment Purposes Only

When Window mode is enabled, the watchdog must be refreshed after the counter has reached a minimum expected time value; otherwise, the watchdog resets the MCU. The minimum expected time value is specified in the WIN register. Setting CS[WIN] enables Window mode.

27.4.1.2 Refreshing the Watchdog

The refresh write sequence can be

- either two 16-bit writes (0xA602, 0xB480)
- or one 32-bit write (0xB480_A602)

to the CNT register. Both methods must occur before the WDG timeout; otherwise, the watchdog resets the MCU.

Note

Before starting the refresh sequence, disable the global interrupts. Otherwise, an interrupt could effectively invalidate the refresh sequence, if the interrupt occurs before the refresh writes finish. After the sequence finishes, restore the global interrupt control state.

27.4.1.3 Example code: Refreshing the Watchdog

The following code segment shows the refresh write sequence of the WDOG module.

```
/* Refresh watchdog */

for (;;) // main loop
{
   ...

   DisableInterrupts; // disable global interrupt


   WDOG_CNT = 0xB480A602 ; // write the 1st and the 2nd refresh words

   EnableInterrupts; // enable global interrupt

   ...
}
```

For Assessment Purposes Only

27.4.2 Configuring the Watchdog

All watchdog control bits, timeout value, and window value are write-once after reset. This means that after a write has occurred they cannot be changed unless a reset occurs. This provides a robust mechanism to configure the watchdog and ensure that a runaway condition cannot mistakenly disable or modify the watchdog configuration after configured.

This is guaranteed by the user configuring the window and timeout value first, followed by the other control bits, and ensuring that CS[UPDATE] is also set to 0. The new configuration takes effect only after all registers except CNT are written once after reset. Otherwise, the WDOG uses the reset values by default. If window mode is not used (CS[WIN] is 0), writing to WIN is not required to make the new configuration take effect.

27.4.2.1 Reconfiguring the Watchdog

In some cases (like when supporting a bootloader function), you may want to reconfigure or disable the watchdog, *without forcing a reset first*.

- By setting CS[UPDATE] to 1 on the initial configuration of the watchdog after a reset, you can reconfigure the watchdog at any time by executing an unlock sequence.
- Conversely, if CS[UPDATE] remains 0, the only way to reconfigure the watchdog is by initiating a reset.

The unlock sequence is similar to the refresh sequence but uses different values.

27.4.2.2 Unlocking the Watchdog

The unlock sequence is a write to the CNT register of 0xC520 followed by 0xD928 within 16 bus clocks at any time after the watchdog has been configured. On completing the unlock sequence, the user must reconfigure the watchdog within 128 bus clocks; otherwise, the watchdog closes the unlock window.

NOTE

Due to the 128 bus clocks requirement for reconfiguring the watchdog, some delays must be inserted before executing STOP or WAIT instructions after reconfiguring the watchdog. This ensures that the watchdog's new configuration takes effect before the MCU enters low power mode. Otherwise, the MCU may not be waken up from low power mode.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

27.4.2.3 Example code: Reconfiguring the Watchdog

The following code segment shows an example reconfiguration of the WDOG module.

```
/* Initialize watchdog with ~1-kHz clock source, ~1s time-out */
DisableInterrupts; // disable global interrupt

while(WDOG_CS[ULK] == 1); // Wait for close unlock window
WDOG_CNT = 0xD928C520; // write the 1st and the 2nd unlock words
while(WDOG_CS[ULK] == 0); // Wait for wdog unlock
WDOG_TOVAL = 1000; // setting timeout value
WDOG_CS = WDOG_CS_CLK_MASK; // setting 1-kHz clock source
WDOG_CS = WDOG_CS_EN_MASK; // enable counter running
while(WDOG_CS_RCS == 0); // wait for wdog completing reconfiguration

EnableInterrupts; // enable global interrupt
```

27.4.3 Clock source

The watchdog counter has the following clock source options selected by programming CS[CLK]:

- bus clock
- internal Low-Power Oscillator clock (LPO_CLK) (This is the default source.)
- internal 8 MHz clock (SIRC)
- external clock (SOSC)

The options allow software to select a clock source independent of the bus clock for applications that need to meet more robust safety requirements. Using a clock source other than the bus clock ensures that the watchdog counter continues to run if the bus clock is somehow halted; see Backup reset.

An optional fixed prescaler for all clock sources allows for longer timeout periods. When CS[PRES] is set, the clock source is prescaled by 256 before clocking the watchdog counter.

The following table summarizes the different watchdog timeout periods available.

For Assessment Purposes Only

Table 27-2. Watchdog timeout availability

Reference clock	Prescaler	Watchdog time-out availability
Internal LPO_CLK	Pass through	~1 ms–65.5 s (if LPO_CLK = 1 kHz); (~1 ms–65.5 s)/128 (if LPO_CLK = 128 kHz). [1]
	÷256	~256 ms–16,777.2 s (if LPO_CLK = 1 kHz); ~2 ms–131.1 s (if LPO_CLK = 128 kHz).
Internal 8 MHz (SIRC)	Pass through	125 ns–8.1925 ms
	÷256	32 µs–2.09728 s
1 MHz (from bus or external)	Pass through	1 µs–65.54 ms
	÷256	256 µs–16.777 s
20 MHz (from bus or external)	Pass through	50 ns–3.277 ms
	÷256	12.8 µs–838.8 ms

1. The default timeout value after reset is approximately 4 ms (if LPO_CLK = 1 kHz), or 4/128 ms (if LPO_CLK = 128 kHz).

NOTE

When the programmer switches clock sources during reconfiguration, the watchdog hardware holds the counter at zero for 2.5 periods of the previous clock source and 2.5 periods of the new clock source after the configuration time period (128 bus clocks) ends. This delay ensures a smooth transition before restarting the counter with the new configuration.

27.4.4 Using interrupts to delay resets

- **When interrupts are enabled (CS[INT] = 1):** After a reset-triggering event (like a counter timeout or invalid refresh attempt), the watchdog first generates an interrupt request. Next, the watchdog delays 128 bus clocks (from the interrupt vector fetch, not the reset-triggering event) before forcing a reset, to allow the interrupt service routine (ISR) to perform tasks (like analyzing the stack to debug code).
- **When interrupts are disabled (CS[INT] = 0):** the watchdog does not delay the forcing a reset.

27.4.5 Backup reset

NOTE

A clock source other than the bus clock must be used as the reference clock for the counter; otherwise, the backup reset function is not available.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The backup reset function is a safeguard feature that independently generates a reset in case the main WDOG logic loses its clock (the bus clock) and can no longer monitor the counter. If the watchdog counter overflows twice in succession (without an intervening reset), the backup reset function takes effect and generates a reset.

27.4.6 Functionality in debug and low-power modes

By default, the watchdog is not functional in Active Background mode, Wait mode, or Stop mode. However, the watchdog can remain functional in these modes as follows:

- **For Active Background mode,** set CS[DBG]. (This way the watchdog is functional in Active Background mode even when the CPU is held by the Debug module.)
- **For Wait mode,** set CS[WAIT].
- **For Stop mode,** set CS[STOP], CS[WAIT], and ensure the clock source is active in STOP mode.

NOTE

The watchdog can not generate interrupt in Stop mode even if CS[STOP] is set and will not wake the MCU from Stop mode. It can generate reset during Stop mode.

For Active Background mode and Stop mode, in addition to the above configurations, a clock source other than the bus clock must be used as the reference clock for the counter; otherwise, the watchdog cannot function.

27.4.7 Fast testing of the watchdog

Before executing application code in safety critical applications, users are required to test that the watchdog works as expected and resets the MCU. Testing every bit of a 16-bit counter by letting it run to the overflow value takes a relatively long time (64 kHz clocks).

To help minimize the startup delay for application code after reset, the watchdog has a feature to test the watchdog more quickly by splitting the counter into its constituent byte-wide stages. The low and high bytes are run independently and tested for timeout against the corresponding byte of the timeout value register. (For complete coverage when testing the high byte of the counter, the test feature feeds the input clock via the 8th bit of the low byte, thus ensuring that the overflow connection from the low byte to the high byte is tested.)

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Using this test feature reduces the test time to 512 clocks (not including overhead, such as user configuration and reset vector fetches). To further speed testing, use a faster clock (such as the bus clock) for the counter reference.

On a power-on reset, the POR bit in the system reset register is set, indicating the user should perform the WDOG fast test.

27.4.7.1 Testing each byte of the counter

The test procedure follows these steps:

1. Program the preferred watchdog timeout value in the TOVAL register during the watchdog configuration time period.
2. Select a byte of the counter to test using the CS[TST] = 10b for the low byte; CS[TST] = 11b for the high byte.
3. Wait for the watchdog to timeout. Optionally, in the idle loop, increment RAM locations as a parallel software counter for later comparison. Because the RAM is not affected by a watchdog reset, the timeout period of the watchdog counter can be compared with the software counter to verify the timeout period has occurred as expected.
4. The watchdog counter times out and forces a reset.
5. Confirm the WDOG flag in the system reset register is set, indicating that the watchdog caused the reset. (The POR flag remains clear.)
6. Confirm that CS[TST] shows a test (10b or 11b) was performed.

If confirmed, the count and compare functions work for the selected byte. Repeat the procedure, selecting the other byte in step 2.

NOTE

CS[TST] is cleared by a POR only and not affected by other resets.

27.4.7.2 Entering user mode

After successfully testing the low and high bytes of the watchdog counter, the user can configure CS[TST] to 01b to indicate the watchdog is ready for use in application user mode. Thus if a reset occurs again, software can recognize the reset trigger as a real watchdog reset caused by runaway or faulty application code.

As an ongoing test when using the default LPO clock source, software can periodically read the CNT register to ensure the counter is being incremented.

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 28
System Clock Generator (SCG)

28.1 Chip-specific System Clock Generator (SCG) information

28.1.1 Information of SCG on this device

The SCG_SOSCCFG[11:8] bitfields are *Reserved* on this device, i.e. SC2P/SC4P/SC8P/SC16P is not appliable, as there is no integrated capacitor in 32k crystal oscillator on this device.

28.2 Introduction

The system clock generator (SCG) module provides the system clocks of the MCU. The SCG contains a phase-locked loop (SPLL) , a slow internal reference clock (SIRC), a fast internal reference clock (FIRC), and the system oscillator clock (SOSC). The PLLs is sourced by the SOSC reference clock. The SCG can select either the output clock of the SPLL or a SCG reference clock (SIRC, FIRC, and SOSC) as the source for the MCU system clocks. The SCG also supports operation with crystal oscillators, which allows an external crystal, ceramic resonator, or another external clock source to produce the external reference clock (which are also available as clock sources for the MCU systems clocks).

28.2.1 Features

Key features of the SCG module are:

- System Phase-locked loop (SPLL):
  - Voltage-controlled oscillator (VCO)

For Assessment Purposes Only

- Selectable Internal or External reference clock is used as the PLL source
- Modulo VCO frequency divider
- Phase/Frequency detector
- Integrated loop filter
- Can be selected as the clock source for the MCU system clocks
- 2 programmable post-dividers clock outputs, which can be used as clock sources for other on-chip peripherals

- 2 Internal reference clock (IRC) generators:
  - Slow IRC clock with programmable High and Low frequency range, with each range having a set of 8 trim bits for accuracy
  - Fast IRC clock with programmable High and Low frequency range, with 3 sets of trim bits for accuracy
  - Either the slow or the fast clock can be selected as the clock source for the MCU system clocks
  - 2 programmable post-divider clock outputs for each IRC, which can be used as clock sources for other on-chip peripherals
- System Crystal Oscillator:
  - Can be used as a source for the System PLL
  - Can be selected as the clock source for the MCU system clocks
- Clock monitor with reset and interrupt request capability for SPLL, SOSC, clocks
- Lock detector with interrupt request capability for use with the SPLL
- Each of the clock sources have reference dividers for clocking on-chip modules and peripherals, namely:
  - SPLLDIV1_CLK / SPLLDIV2_CLK
  - FIRCDIV1_CLK / SCG_FIRCDIV2_CLK
  - SIRCDIV1_CLK / SIRCDIV2_CLK
  - SOSCDIV1_CLK / SOSCDIV2_CLK

See the Clock Distribution Chapter for more information.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

28.3 Memory Map/Register Definition

This section includes the memory map and register definition.

The SCG registers can only be written when in supervisor mode. Write accesses when in user mode will result in a bus transfer error. Read accesses may be performed in both supervisor and user mode.

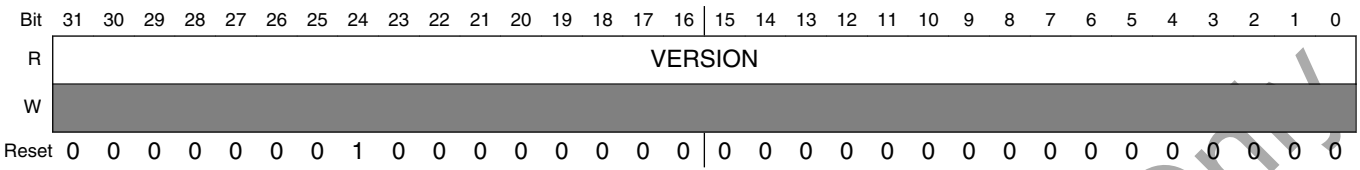
SCG memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Version ID Register (SCG_VERID)	32	R	0100_0000h	28.3.1/476
4	Parameter Register (SCG_PARAM)	32	R	See section	28.3.2/476
10	Clock Status Register (SCG_CSR)	32	R	See section	28.3.3/477
14	Run Clock Control Register (SCG_RCCR)	32	R/W	See section	28.3.4/480
18	VLPR Clock Control Register (SCG_VCCR)	32	R/W	See section	28.3.5/482
1C	HSRUN Clock Control Register (SCG_HCCR)	32	R/W	See section	28.3.6/485
20	SCG CLKOUT Configuration Register (SCG_CLKOUTCNFG)	32	R/W	0300_0000h	28.3.7/487
100	System OSC Control Status Register (SCG_SOSCCSR)	32	R/W	See section	28.3.8/488
104	System OSC Divide Register (SCG_SOSCDIV)	32	R/W	0000_0000h	28.3.9/490
108	System Oscillator Configuration Register (SCG_SOSCCFG)	32	R/W	0000_0010h	28.3.10/491
200	Slow IRC Control Status Register (SCG_SIRCCSR)	32	R/W	0100_0005h	28.3.11/493
204	Slow IRC Divide Register (SCG_SIRCDIV)	32	R/W	0000_0000h	28.3.12/494
208	Slow IRC Configuration Register (SCG_SIRCCFG)	32	R/W	0000_0001h	28.3.13/495
300	Fast IRC Control Status Register (SCG_FIRCCSR)	32	R/W	See section	28.3.14/496
304	Fast IRC Divide Register (SCG_FIRCDIV)	32	R/W	0000_0000h	28.3.15/498
308	Fast IRC Configuration Register (SCG_FIRCCFG)	32	R/W	0000_0000h	28.3.16/499
30C	Fast IRC Trim Configuration Register (SCG_FIRCTCFG)	32	R/W	0000_0000h	28.3.17/499
318	Fast IRC Status Register (SCG_FIRCSTAT)	32	R	See section	28.3.18/501
600	System PLL Control Status Register (SCG_SPLLCSR)	32	R/W	0000_0000h	28.3.19/502
604	System PLL Divide Register (SCG_SPLLDIV)	32	R/W	0000_0000h	28.3.20/504
608	System PLL Configuration Register (SCG_SPLLCFG)	32	R/W	0000_0000h	28.3.21/505

For Review Only

28.3.1 Version ID Register (SCG_VERID)

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

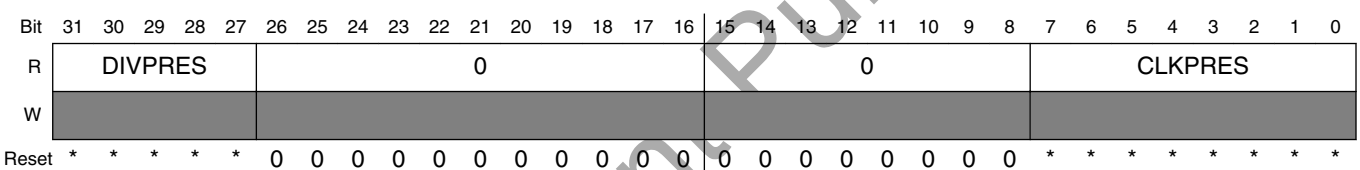


SCG_VERID field descriptions

Field	Description
VERSION	SCG Version Number

28.3.2 Parameter Register (SCG_PARAM)

Address: 0h base + 4h offset = 4h



* Notes:
- DIVPRES field: The reset value is controlled by which SCG System Dividers are used by Soc.
- CLKPRES field: The reset value is controlled by which SCG Clock Sources are used by Soc. Please reference the Reference manual clocking chapter.

SCG_PARAM field descriptions

Field	Description
31–27 DIVPRES	Divider Present Indicates which system clock dividers are present in this instance of SCG. DIVPRES[27]=1 System DIVSLOW is present. DIVPRES[28]=1 System DIVBUS is present DIVPRES[30]=1 System DIVPLAT is present DIVPRES[31]=1 System DIVCORE is present
26–16 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.
CLKPRES	Clock Present

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_PARAM field descriptions (continued)

Field	Description
	Indicates which clock sources are present in this instance of SCG. Any bits not defined in this bit field are Reserved and always has the value 0 when read. CLKPRES[0] Reserved CLKPRES[1]=1 System OSC (SOSC) is present CLKPRES[2]=1 Slow IRC (SIRC) is present CLKPRES[3]=1 Fast IRC (FIRC) is present CLKPRES[6]=1 System PLL (SPLL) is present

28.3.3 Clock Status Register (SCG_CSR)

This register returns the currently configured system clock source and the system clock dividers for the core (DIVCORE) and peripheral interface clock (DIVSLOW). The SCG_CSR reflects the configuration set by one of three clock control registers SCG_RCCR, SCG_VCCR, SCG_HCCR.

Address: 0h base + 10h offset = 10h

Bit	31	30	29	28	27	26	25	24	23	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				SCS				0				DIVCORE				DIVPLAT				0				DIVBUS				DIVSLOW			
W																																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	*	*	*	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

* Notes:

- DIVCORE field: The reset value is controlled by user FOPT bits that get uploaded during reset. The two option reset values are div-by-1 or div-by-2 when resetting into RUN mode or div-by-4 or div-by-8 when resetting into VLPR mode

SCG_CSR 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 SCS	System Clock Source Returns the currently configured clock source generating the system clock. 0000 Reserved 0001 System OSC (SOSC_CLK) 0010 Slow IRC (SIRC_CLK) 0011 Fast IRC (FIRC_CLK) 0100 Reserved 0101 Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_CSR field descriptions (continued)

Field	Description
	0110 System PLL (SPLL_CLK) 0111 Reserved
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 DIVCORE	Core Clock Divide Ratio 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
15–12 DIVPLAT	Platform Clock Divide Ratio 0000 Divide-by-1 0001 Reserved 0010 Reserved 0011 Reserved 0100 Reserved 0101 Reserved 0110 Reserved 0111 Reserved 1000 Reserved 1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved
11–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–4 DIVBUS	Bus Clock Divide Ratio 0000 Divide-by-1 0001 Divide-by-2 0010 Divide-by-3

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_CSR field descriptions (continued)

Field	Description
	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
DIVSLOW	Slow Clock Divide Ratio 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 Reserved 1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved

For Asse

Freescale Confidential Proprietary - Non-Disclosure Agreement required

28.3.4 Run Clock Control Register (SCG_RCCR)

This register controls the system clock source and the system clock dividers for the core, platform, external and bus clock domains when in Run mode only. This register can only be written using a 32-bit write. Selecting a different clock source when in RUN requires that clock source to be enabled first and be valid before system clocks switch to that clock source. If system clock divide ratios also change when selecting a different clock mode when in RUN, new system clock divide ratios will not take affect until new clock source is valid.

Address: 0h base + 14h offset = 14h

Bit	31	30	29	28	27	26	25	24	23	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				SCS				0				DIVCORE				DIVPLAT				0				DIVBUS				DIVSLOW			
W																																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	*	*	*	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

* Notes:
- DIVCORE field: The reset value is controlled by user FOPT bits that get uploaded during reset. The two option reset values are div-by-1 and div-by-2

SCG_RCCR 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 SCS	System Clock Source Selects the clock source generating the system clock in Run mode. Attempting to select a clock that is not valid will be ignored. Selecting a different clock source when in Run mode requires that clock source to be enabled first and be valid before system clocks are allowed to switch to that clock source. 0000 Reserved 0001 System OSC (SOSC_CLK) 0010 Slow IRC (SIRC_CLK) 0011 Fast IRC (FIRC_CLK) 0100 Reserved 0101 Reserved 0110 System PLL (SPLL_CLK) 0111 Reserved
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 DIVCORE	Core Clock Divide Ratio 0000 Divide-by-1 0001 Divide-by-2 0010 Divide-by-3 0011 Divide-by-4

Table continues on the next page...

Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_RCCR field descriptions (continued)

Field	Description
	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
15–12 DIVPLAT	Platform Clock Divide Ratio 0000 Divide-by-1 0001 Reserved 0010 Reserved 0011 Reserved 0100 Reserved 0101 Reserved 0110 Reserved 0111 Reserved 1000 Reserved 1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved
11–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–4 DIVBUS	Bus Clock Divide Ratio 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

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_RCCR field descriptions (continued)

Field	Description
	1101 Divide-by-14 1110 Divide-by-15 1111 Divide-by-16
DIVSLOW	Slow Clock Divide Ratio 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 Reserved 1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved

28.3.5 VLPR Clock Control Register (SCG_VCCR)

This register controls the system clock source and the system clock dividers for the core, platform, external and bus clock domains when in VLPR mode only. This register can only be written using a 32-bit write. Selecting a different clock source when in VLPR requires that clock source to be enabled first and be valid before system clocks switch to that clock source. If system clock divide ratios also change when selecting a different clock mode when in VLPR, new system clock divide ratios will not take affect until new clock source is valid.

Address: 0h base + 18h offset = 18h

Bit	31	30	29	28	27	26	25	24	23	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				SCS				0				DIVCORE				DIVPLAT				0				DIVBUS				DIVSLOW			
W																																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	*	*	*	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

* Notes:

- DIVCORE field: The reset value is controlled by user FOPT bits that get uploaded during reset. The two option reset values are div-by-4 and div-by-8.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_VCCR 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 SCS	System Clock Source Selects the clock source generating the system clock in VLPR mode. Attempting to select a clock that is not valid will be ignored. Selects the clock source generating the system clock. Selecting a different clock source when in VLPR mode requires that clock source to be enabled first and be valid before system clocks switch to that clock source. 0000 Reserved 0001 System OSC (SOSC_CLK) 0010 Slow IRC (SIRC_CLK) 0011 Reserved 0100 Reserved 0101 Reserved 0110 Reserved 0111 Reserved
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 DIVCORE	Core Clock Divide Ratio 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
15–12 DIVPLAT	Platform Clock Divide Ratio 0000 Divide-by-1 0001 Reserved 0010 Reserved 0011 Reserved 0100 Reserved 0101 Reserved 0110 Reserved 0111 Reserved 1000 Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_VCCR field descriptions (continued)

Field	Description
	1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved
11–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–4 DIVBUS	Bus Clock Divide Ratio 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
DIVSLOW	Slow Clock Divide Ratio 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 Reserved 1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved

Freescale Confidential Proprietary - Non-Disclosure Agreement required

28.3.6 HSRUN Clock Control Register (SCG_HCCR)

This register controls the system clock source and the system clock dividers for the core, platform, external and bus clock domains when in HSRUN mode only. This register can only be written using a 32-bit write. Selecting a different clock source when in HSRUN requires that clock source to be enabled first and be valid before system clocks switch to that clock source. If system clock divide ratios also change when selecting a different clock mode when in HSRUN, new system clock divide ratios will not take affect until new clock source is valid.

Address: 0h base + 1Ch offset = 1Ch

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

SCG_HCCR 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 SCS	System Clock Source Selects the clock source generating the system clock in HSRUN mode. Attempting to select a clock that is not valid will be ignored. Selecting a different clock source when in HSRUN mode will enable that clock source and switch to that clock mode when it is valid. 0000 Reserved 0001 System OSC (SOSC_CLK) 0010 Slow IRC (SIRC_CLK) 0011 Fast IRC (FIRC_CLK) 0100 Reserved 0101 Reserved 0110 System PLL (SPLL_CLK) 0111 Reserved
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 DIVCORE	Core Clock Divide Ratio 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

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_HCCR field descriptions (continued)

Field	Description
	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
15–12 DIVPLAT	Platform Clock Divide Ratio 0000 Divide-by-1 0001 Reserved 0010 Reserved 0011 Reserved 0100 Reserved 0101 Reserved 0110 Reserved 0111 Reserved 1000 Reserved 1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved
11–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–4 DIVBUS	Bus Clock Divide Ratio 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
DIVSLOW	Slow Clock Divide Ratio

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

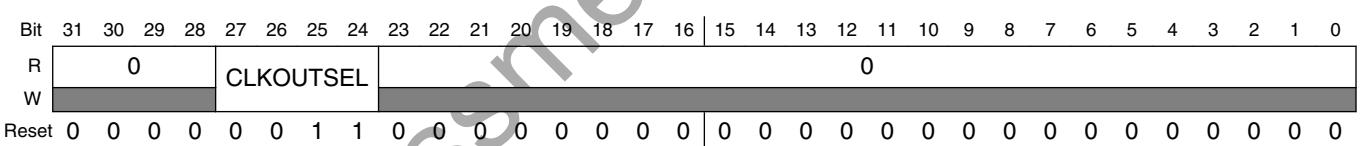
SCG_HCCR field descriptions (continued)

Field	Description
	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 Reserved 1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved

28.3.7 SCG CLKOUT Configuration Register (SCG_CLKOUTCNFG)

This register controls which SCG clock source is selected to be ported out to the CLKOUT pin.

Address: 0h base + 20h offset = 20h



SCG_CLKOUTCNFG 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 CLKOUTSEL	SCG Clkout Select Selects the SCG system clock. 0000 SCG SLOW Clock 0001 System OSC (SOSC_CLK) 0010 Slow IRC (SIRC_CLK) 0011 Fast IRC (FIRC_CLK) 0100 Reserved 0101 Reserved 0110 System PLL (SPLL_CLK)

Table continues on the next page...

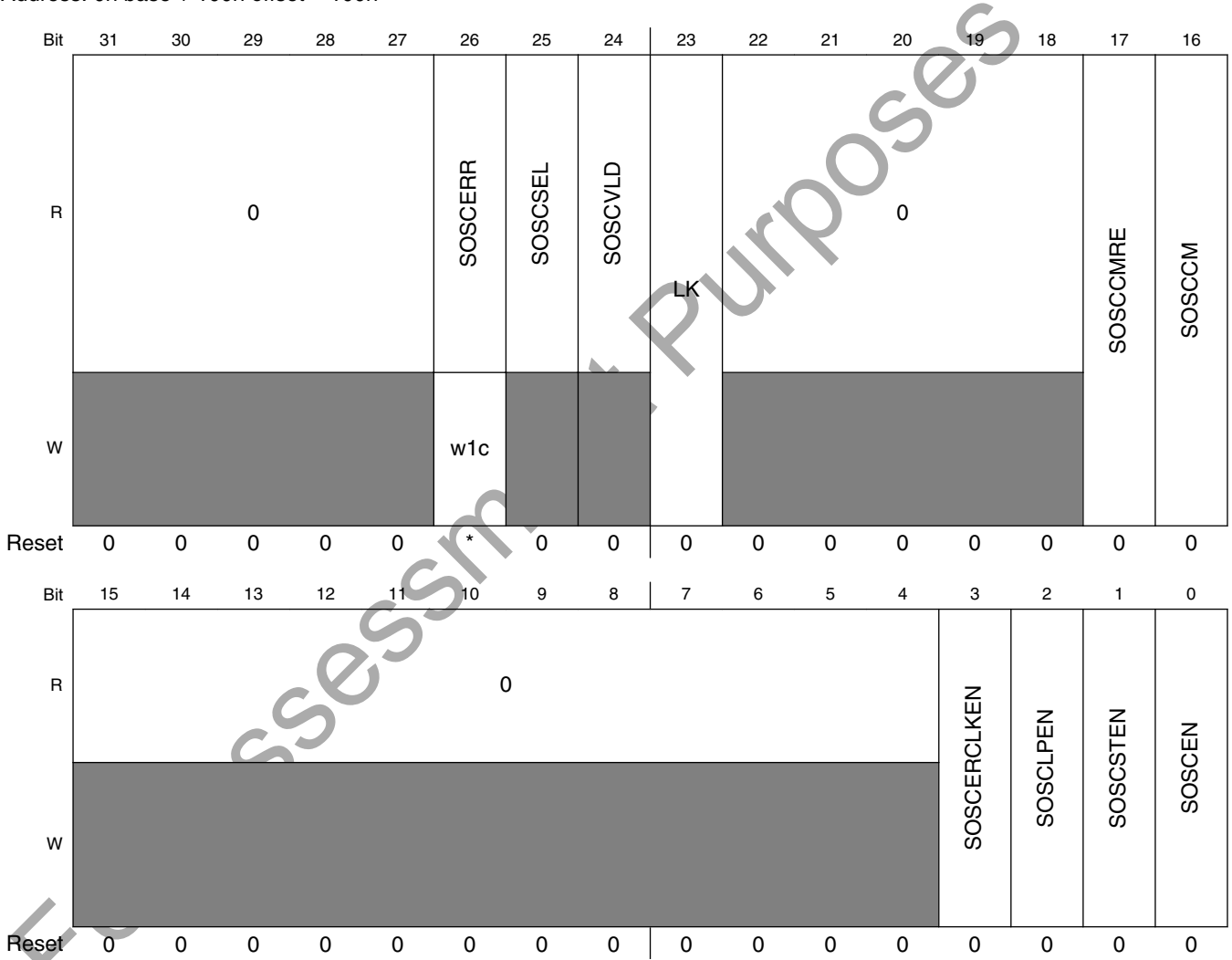
Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_CLKOUTCNFG field descriptions (continued)

Field	Description
	0111 Reserved 1111 Reserved
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

28.3.8 System OSC Control Status Register (SCG_SOSCCSR)

Address: 0h base + 100h offset = 100h



* Notes:

- SOSCERR field: This flag is reset on Chip POR only

Freescale Confidential Proprietary Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_SOSCCSR field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 SOSCERR	System OSC Clock Error This flag is reset on Chip POR only, software can also clear this flag by writing a logic one. 0 System OSC Clock Monitor is disabled or has not detected an error 1 System OSC Clock Monitor is enabled and detected an error
25 SOSCSEL	System OSC Selected 0 System OSC is not the system clock source 1 System OSC is the system clock source
24 SOSCVLD	System OSC Valid 0 System OSC is not enabled or clock is not valid 1 System OSC is enabled and output clock is valid
23 LK	Lock Register 0 This Control Status Register can be written. 1 This Control Status Register cannot be written.
22–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 SOSCCMRE	System OSC Clock Monitor Reset Enable 0 Clock Monitor generates interrupt when error detected 1 Clock Monitor generates reset when error detected
16 SOSCCM	System OSC Clock Monitor Enables the clock monitor when SOSCVLD is set. If the clock source is disabled in a low power mode then the clock monitor is also disabled in the low power mode. When the clock monitor is disabled in a low power mode, it remains disabled until the clock valid flag is set following exit from the low power mode. 0 System OSC Clock Monitor is disabled 1 System OSC Clock Monitor is enabled
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 SOSCERCLKEN	System OSC 3V ERCLK Enable SOSCERCLKEN is required for stop modes. 0 System OSC 3V ERCLK output clock is disabled. 1 System OSC 3V ERCLK output clock is enabled when SYSOSC is enabled.
2 SOSCLPEN	System OSC Low Power Enable SOSCLPEN is required for low power modes. In VLPS mode (low power stop mode), if you want the clock to remain ON, then both SOSCLPEN and SOSCSTEN bits must be enabled. 0 System OSC is disabled in VLP modes 1 System OSC is enabled in VLP modes

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_SOSCCSR field descriptions (continued)

Field	Description
1 SOSCSTEN	System OSC Stop Enable 0 System OSC is disabled in Stop modes 1 System OSC is enabled in Stop modes if SOSCEN=1.
0 SOSCEN	System OSC Enable 0 System OSC is disabled 1 System OSC is enabled

28.3.9 System OSC Divide Register (SCG_SOSCDIV)

The SCG_SOSCDIV register provides the control of 3 clock trees which can be used to provide optional peripheral functional clocks, or alternative module clocks. Each clock tree has optional dividers of the input SOSC clock. Changes to SOSCDIV should be done when System OSC is disabled to prevent glitches to output divided clock.

Address: 0h base + 104h offset = 104h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					SOSCDIV2			0					SOSCDIV1		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SCG_SOSCDIV field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 SOSCDIV2	System OSC Clock Divide 2 Clock divider 2 for System OSC. 000 Output disabled 001 Divide by 1 010 Divide by 2 011 Divide by 4 100 Divide by 8 101 Divide by 16

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

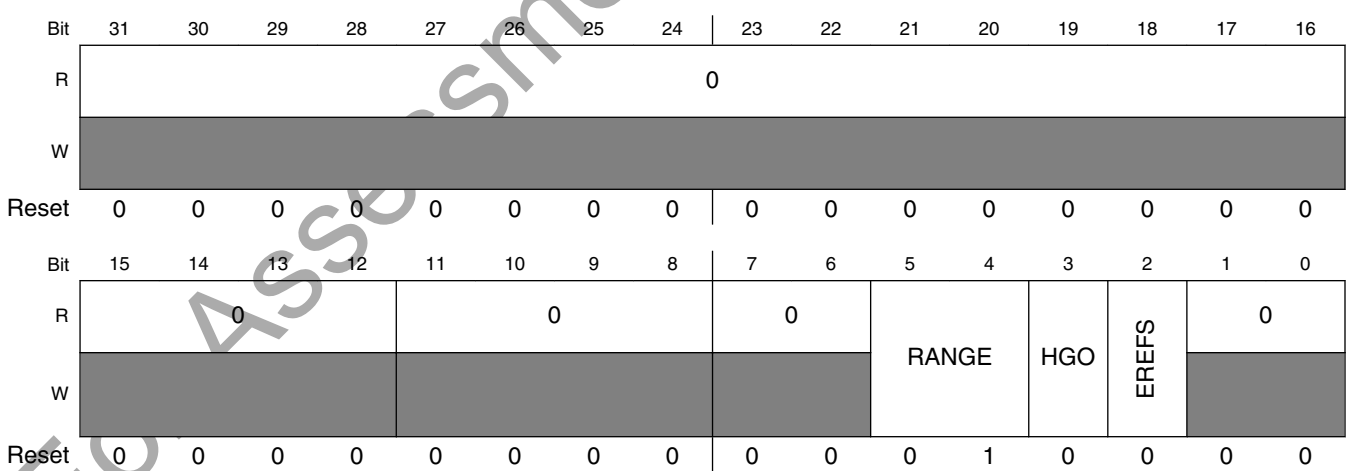
SCG_SOSCDIV field descriptions (continued)

Field	Description
	110 Divide by 32 111 Divide by 64
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SOSCDIV1	System OSC Clock Divide 1 Clock divider 1 for System OSC. Used to generate the system clock source and by platform clock modules that need an asynchronous clock source. 000 Output disabled 001 Divide by 1 010 Divide by 2 011 Divide by 4 100 Divide by 8 101 Divide by 16 110 Divide by 32 111 Divide by 64

28.3.10 System Oscillator Configuration Register (SCG_SOSCCFG)

The SOSCCFG register cannot be changed when the System OSC is enabled. When the System OSC is enabled, writes to this register are ignored, and there is no transfer error.

Address: 0h base + 108h offset = 108h



SCG_SOSCCFG field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_SOSCCFG field descriptions (continued)

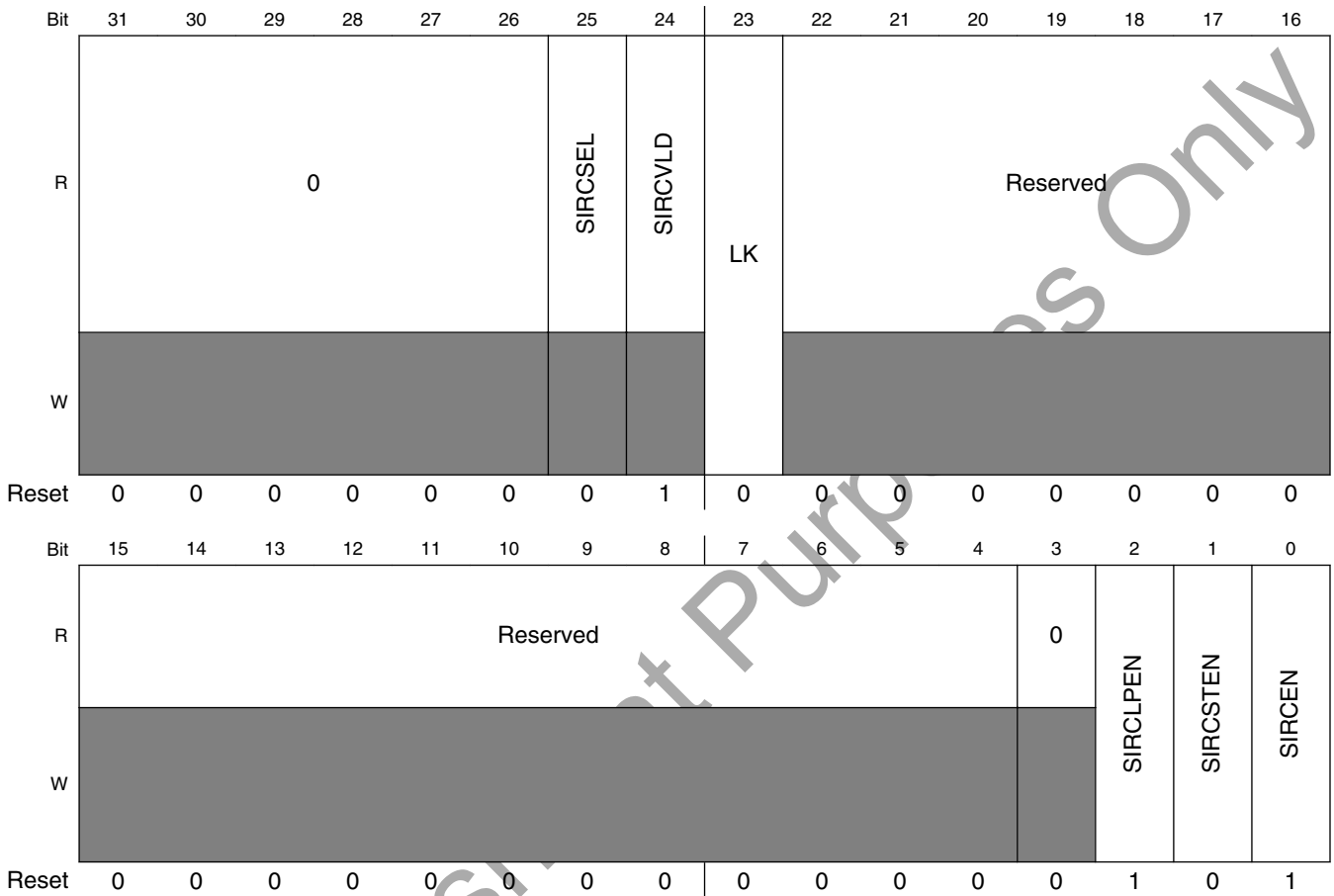
Field	Description
11–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 RANGE	System OSC Range Select Selects the frequency range for the system crystal oscillator (OSC) 00 Reserved 01 Low frequency range selected for the crystal oscillator of 32 kHz to 40 kHz. 10 Medium frequency range selected for the crytstal oscillator of 1 Mhz to 8 Mhz. 11 High frequency range selected for the crystal oscillator of 8 Mhz to 32 Mhz.
3 HGO	High Gain Oscillator Select Controls the crystal oscillator power mode of operations. 0 Configure crystal oscillaor 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. This bit selects which clock is output from the System OSC (SOSC) into the SCG, thus either the crystal oscillator or from an external clock input 0 External reference clock selected 1 Internal crystal oscillator of OSC requested.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

For Assessme

Freescale Confidential Proprietary - Non-Disclosure Agreement required

28.3.11 Slow IRC Control Status Register (SCG_SIRCCSR)

Address: 0h base + 200h offset = 200h



Preliminary Purposes Only

SCG_SIRCCSR field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 SIRCSEL	Slow IRC Selected 0 Slow IRC is not the system clock source 1 Slow IRC is the system clock source
24 SIRCVLD	Slow IRC Valid 0 Slow IRC is not enabled or clock is not valid 1 Slow IRC is enabled and output clock is valid
23 LK	Lock Register 0 Control Status Register can be written. 1 Control Status Register cannot be written.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_SIRCCSR field descriptions (continued)

Field	Description
22–4 Reserved	This field is reserved and is always has the value 0 This field is reserved.
3 Reserved	This field is reserved and is always has the value 0 This field is reserved. This read-only field is reserved and always has the value 0.
2 SIRCLPEN	Slow IRC Low Power Enable 0 Slow IRC is disabled in VLP modes 1 Slow IRC is enabled in VLP modes
1 SIRCSTEN	Slow IRC Stop Enable 0 Slow IRC is disabled in Stop modes 1 Slow IRC is enabled in Stop modes
0 SIRCEN	Slow IRC Enable 0 Slow IRC is disabled 1 Slow IRC is enabled

28.3.12 Slow IRC Divide Register (SCG_SIRCDIV)

To prevent glitches to the output divided clock, change SIRDIV when the Slow IRC is disabled.

Address: 0h base + 204h offset = 204h

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

SCG_SIRCDIV field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 SIRCDIV2	Slow IRC Clock Divide 2 Clock divider 2 for Slow IRC. 000 Output disabled 001 Divide by 1

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

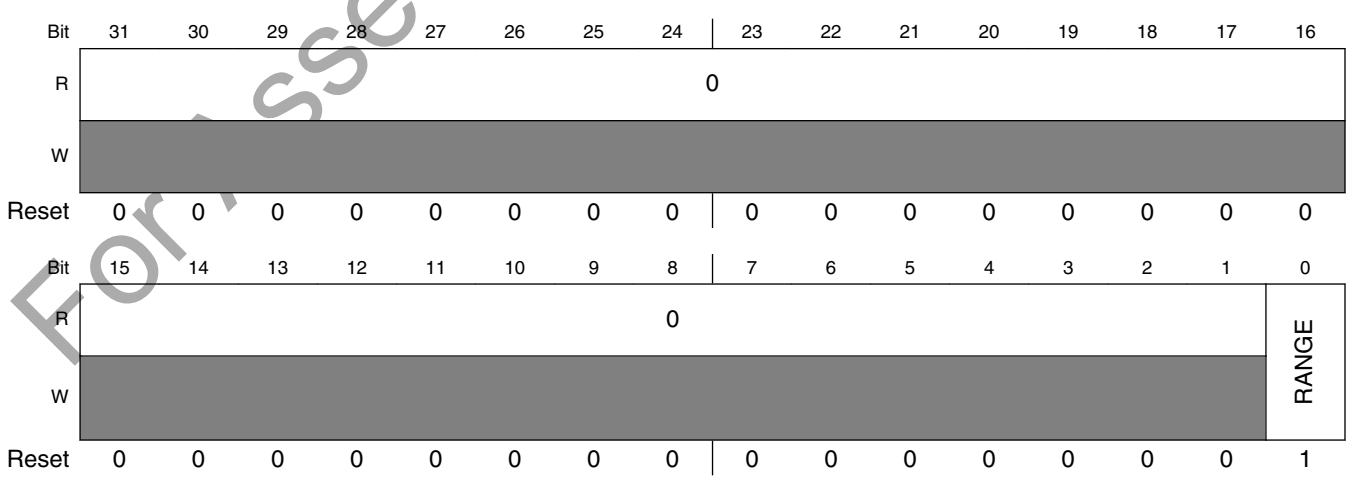
SCG_SIRCDIV field descriptions (continued)

Field	Description
	010 Divide by 2 011 Divide by 4 100 Divide by 8 101 Divide by 16 110 Divide by 32 111 Divide by 64
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SIRCDIV1	Slow IRC Clock Divide 1 Clock divider 1 for Slow IRC. Used to generate the system clock source and by platform clock modules that need an asynchronous clock source. 000 Output disabled 001 Divide by 1 010 Divide by 2 011 Divide by 4 100 Divide by 8 101 Divide by 16 110 Divide by 32 111 Divide by 64

28.3.13 Slow IRC Configuration Register (SCG_SIRCCFG)

The SIRCCFG register cannot be changed when the slow IRC clock is enabled. When the slow IRC clock is enabled, writes to this register are ignored, and there is no transfer error.

Address: 0h base + 208h offset = 208h



For Assessment

SCG_SIRCCFG field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 RANGE	Frequency Range 0 Slow IRC low range clock (2 MHz) 1 Slow IRC high range clock (8 MHz)

28.3.14 Fast IRC Control Status Register (SCG_FIRCCSR)

Address: 0h base + 300h offset = 300h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					FIRCERR	FIRCSEL	FIRCVLD	LK	0						
W						w1c										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						FIRCTRUP	FIRCTREN	0				FIRCREGOFF	FIRCLPEN	FIRCSTEN	FIRCEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Freescale Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_FIRCCSR field descriptions

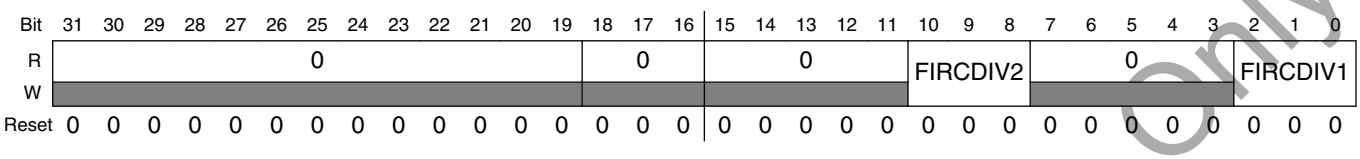
Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 FIRCERR	Fast IRC Clock Error This flag is reset on Chip POR only, software can also clear this flag by writing a logic one 0 Error not detected with the Fast IRC trimming. 1 Error detected with the Fast IRC trimming.
25 FIRCSEL	Fast IRC Selected status 0 Fast IRC is not the system clock source 1 Fast IRC is the system clock source
24 FIRCVLD	Fast IRC Valid status 0 Fast IRC is not enabled or clock is not valid 1 Fast IRC is enabled and output clock is valid
23 LK	Lock Register 0 Control Status Register can be written. 1 Control Status Register cannot be written.
22–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 FIRCTRUP	Fast IRC Trim Update 0 Disable Fast IRC trimming updates 1 Enable Fast IRC trimming updates
8 FIRCTREN	Fast IRC Trim Enable 0 Disable trimming Fast IRC to an external clock source 1 Enable trimming Fast IRC to an external clock source
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 FIRCREGOFF	Fast IRC Regulator Enable 0 Fast IRC Regulator is enabled. 1 Fast IRC Regulator is disabled.
2 FIRCLPEN	Fast IRC Low Power Enable 0 Fast IRC is disabled in VLP modes 1 Fast IRC is enabled in VLP modes
1 FIRCSTEN	Fast IRC Stop Enable 0 Fast IRC is disabled in Stop modes. When selected as the reference clock to the System PLL and if the System PLL is enabled in STOP mode, the Fast IRC will stay enabled even if FIRCSTEN=0. 1 Fast IRC is enabled in Stop modes
0 FIRCEN	Fast IRC Enable 0 Fast IRC is disabled 1 Fast IRC is enabled

Freescale Confidential Proprietary - Non-Disclosure Agreement required

28.3.15 Fast IRC Divide Register (SCG_FIRCDIV)

Changes to FIRCDIV should be done when FAST IRC is disabled to prevent glitches to output divided clock.

Address: 0h base + 304h offset = 304h



Only

SCG_FIRCDIV field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 FIRCDIV2	Fast IRC Clock Divide 2 Clock divider 2 for the Fast IRC. 000 Output disabled 001 Divide by 1 010 Divide by 2 011 Divide by 4 100 Divide by 8 101 Divide by 16 110 Divide by 32 111 Divide by 64
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FIRCDIV1	Fast IRC Clock Divide 1 Clock divider 1 for Fast IRC. Used to generate the system clock source and by platform clock modules that need an asynchronous clock source. 000 Output disabled 001 Divide by 1 010 Divide by 2 011 Divide by 4 100 Divide by 8 101 Divide by 16 110 Divide by 32 111 Divide by 64

Freescale Confidential Proprietary - Non-Disclosure Agreement required

28.3.16 Fast IRC Configuration Register (SCG_FIRCCFG)

The FIRCCFG register cannot be changed when the Fast IRC is enabled. When the Fast IRC is enabled, writes to this register are ignored, and there is no transfer error.

Address: 0h base + 308h offset = 308h

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														RANGE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SCG_FIRCCFG field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RANGE	Frequency Range 00 Fast IRC is trimmed to 48 MHz 01 Fast IRC is trimmed to 52 MHz 10 Fast IRC is trimmed to 56 MHz 11 Fast IRC is trimmed to 60 MHz

28.3.17 Fast IRC Trim Configuration Register (SCG_FIRCTCFG)

The FIRCTCFG register cannot be changed when Fast IRC tuning is enabled. When the Fast IRC tuning is enabled, writes to this register are ignored, and there is no transfer error.

Address: 0h base + 30Ch offset = 30Ch

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					TRIMDIV			0						TRIMSRC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

For Assessment Purposes Only

SCG_FIRCTCFG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 TRIMDIV	Fast IRC Trim Predivide Divide the System OSC down for Fast IRC trimming. 000 Divide by 1 001 Divide by 128 010 Divide by 256 011 Divide by 512 100 Divide by 1024 101 Divide by 2048 110 Reserved. Writing this value will result in Divide by 1. 111 Reserved. Writing this value will result in a Divide by 1.
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TRIMSRC	Trim Source Configures the external clock source to tune the Fast IRC. 00 Reserved 01 Reserved 10 System OSC 11 Reserved

For Assessme

Freescale Confidential Proprietary - Non-Disclosure Agreement required

28.3.18 Fast IRC Status Register (SCG_FIRCSTAT)

This register is loaded from IFR during reset. These register gets uploaded with the trim values generated by FIRC auto trimming which is enabled when FIRC is enabled and FIRCTREN=1 and FIRCTRUP=1. When FIRC auto trimming is enabled and FIRCTRUP is off, writes to this register is allowed and values written to this register are used to trim FIRC clock.

Address: 0h base + 318h offset = 318h

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		TRIMCOAR						0	TRIMFINE						
W																
Reset	0	0	*	*	*	*	*	*	0	*	*	*	*	*	*	*

* Notes:
- TRIMCOAR field: Reset values are loaded out of IFR.
- TRIMFINE field: Reset values are loaded out of IFR.

Purposes Only

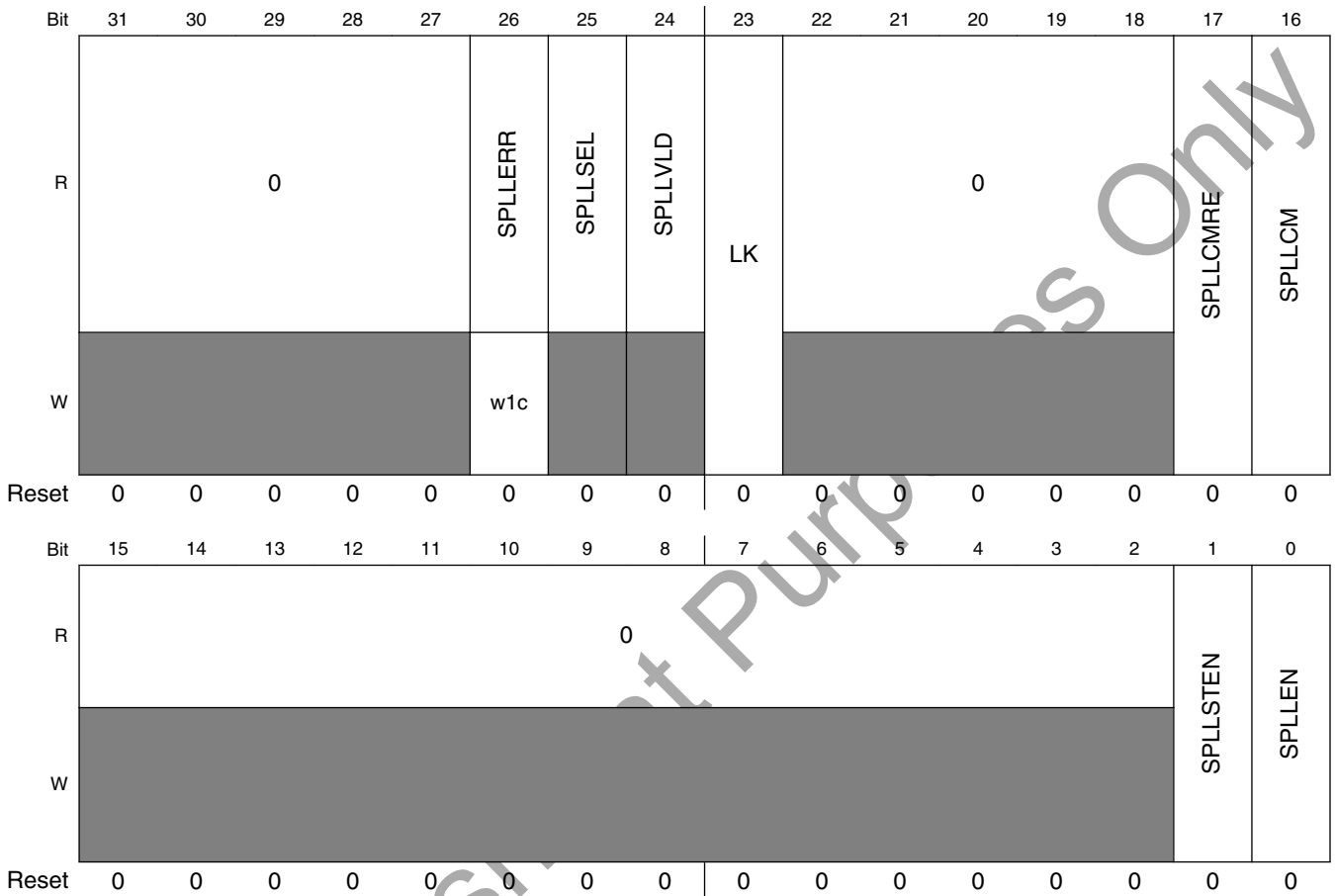
SCG_FIRCSTAT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–8 TRIMCOAR	Trim Coarse TRIMCOAR bits are used to coursely trim the Fast IRC Clock to within approximately 0.7% of the target frequency. When FIRC is enabled and auto trimming is enabled (FIRCTREN=1 and FIRCTRUP=1), then TRIMCOAR register gets uploaded with the trimmed coarse value. When FIRCTRUP=0, TRIMCOAR register is writable, to allow user programming of coarse trim values.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TRIMFINE	Trim Fine Status Once the Fast IRC Clock is trimmed to 0.7% of the target frequency using the TRIMCOAR bits, the TRIMFINE bits can be used to trim the Fast IRC Clock to within 0.04% of the target frequency. When FIRC is enabled and auto trimming is enabled (FIRCTREN=1 and FIRCTRUP=1), TRIMFINE register gets uploaded with the trimmed fine value. When FIRCTRUP=0, TRIMFINE register is writeable, to allow user programming of fine trim values.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

28.3.19 System PLL Control Status Register (SCG_SPLLCSR)

Address: 0h base + 600h offset = 600h



SCG_SPLLCSR field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 SPLLERR	System PLL Clock Error This flag is reset on Chip POR only, software can also clear this flag by writing a logic one 0 System PLL Clock Monitor is disabled or has not detected an error 1 System PLL Clock Monitor is enabled and detected an error. System PLL Clock Error flag will not set when System OSC is selected as its source and SOSCERR has set.
25 SPLLSEL	System PLL Selected 0 System PLL is not the system clock source 1 System PLL is the system clock source
24 SPLLVLD	System PLL Valid

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG_SPLLCSR field descriptions (continued)

Field	Description
	Indicates when the SPLL clock is valid. Disabling the SPLL or a SOSC error when selected as the reference clock to the SPLL will cause the SPLLVLD to clear without setting SPLLERROR. 0 System PLL is not enabled or clock is not valid 1 System PLL is enabled and output clock is valid
23 LK	Lock Register 0 Control Status Register can be written. 1 Control Status Register cannot be written.
22–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 SPLLCMRE	System PLL Clock Monitor Reset Enable 0 Clock Monitor generates interrupt when error detected 1 Clock Monitor generates reset when error detected
16 SPLLCM	System PLL Clock Monitor Enables the clock monitor, if the clock source is disabled in a low power mode then the clock monitor is also disabled in the low power mode. When the clock monitor is disabled in a low power mode, it remains disabled until the clock valid flag is set following exit from the low power mode. 0 System PLL Clock Monitor is disabled 1 System PLL Clock Monitor is enabled
15–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 SPLLSTEN	System PLL Stop Enable 0 System PLL is disabled in Stop modes 1 System PLL is enabled in Stop modes
0 SPLLEN	System PLL Enable 0 System PLL is disabled 1 System PLL is enabled

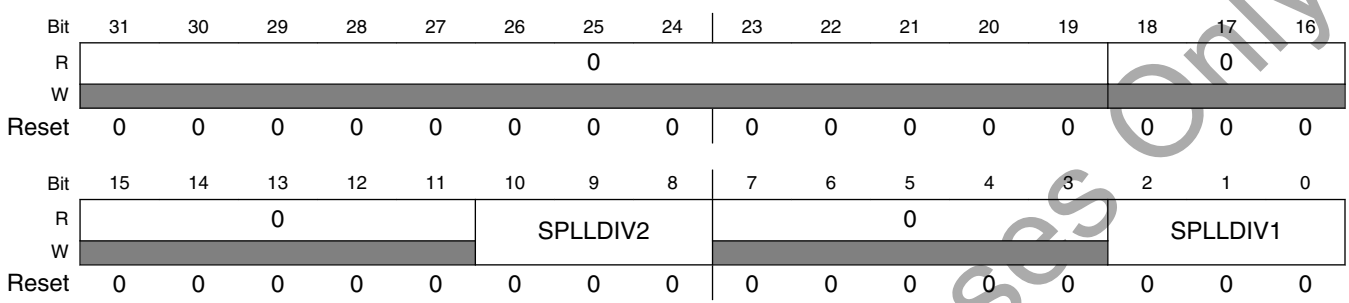
For Asse

Freescale Confidential Proprietary - Non-Disclosure Agreement required

28.3.20 System PLL Divide Register (SCG_SPLLDIV)

Changes to SPLLDIV should be done when System PLL is disabled to prevent glitches to output divided clock.

Address: 0h base + 604h offset = 604h



Freescale Confidential Proprietary Use Only

SCG_SPLLDIV field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 SPLLDIV2	System PLL Clock Divide 2 Clock divider 2 for System PLL. 000 Clock disabled 001 Divide by 1 010 Divide by 2 011 Divide by 4 100 Divide by 8 101 Divide by 16 110 Divide by 32 111 Divide by 64
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SPLLDIV1	System PLL Clock Divide 1 Clock divider 1 for System PLL. Used to generate the system clock source used by platform clock modules that need an asynchronous clock source. 000 Clock disabled 001 Divide by 1 010 Divide by 2 011 Divide by 4 100 Divide by 8

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

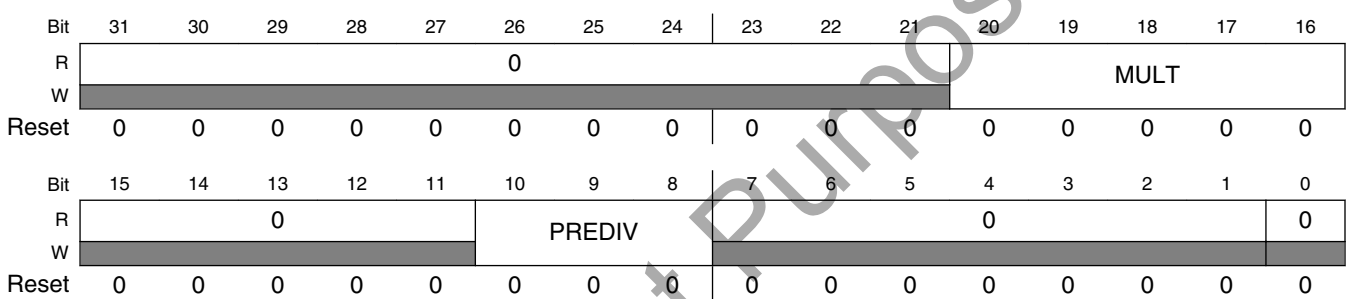
SCG_SPLLDIV field descriptions (continued)

Field	Description
	101 Divide by 16 110 Divide by 32 111 Divide by 64

28.3.21 System PLL Configuration Register (SCG_SPLLCFG)

The SPLLCFG register cannot be changed when the System PLL is enabled. When the System PLL is enabled, writes to this register are ignored, and there is no transfer error. The below information applies to VCO_CLK_2x. The VCO_CLK = (VCO_CLK_2x)/2.

Address: 0h base + 608h offset = 608h



SCG_SPLLCFG field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 MULT	System PLL Multiplier Multiplier for the System PLL. The MULT bits establish the multiplication factor applied to the PLL reference clock frequency.

Table 28-1. PLL VCO Multiply Factor

MULT	Multiply Factor		MULT	Multiply Factor		MULT	Multiply Factor		MULT	Multiply Factor
00000	16		01000	24		10000	32		11000	40
00001	17		01001	25		10001	33		11001	41
00010	18		01010	26		10010	34		11010	42
00011	19		01011	27		10011	35		11011	43
00100	20		01100	28		10100	36		11100	44
00101	21		01101	29		10101	37		11101	45
00110	22		01110	30		10110	38		11110	46
00111	23		01111	31		10111	39		11111	47

Table continues on the next page...

Content Purposes Only

SCG_SPLLCFG field descriptions (continued)

Field	Description
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 PREDIV	PLL Reference Clock Divider Selects the amount to divide down the reference clock for the System PLL. The resulting frequency must be in the range of 8 MHz to 32 MHz.
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table 28-2. System PLL Reference Divide Factor

PREDIV	Divide Factor
000	1
001	2
010	3
011	4
100	5
101	6
110	7
111	8

28.4 Functional description

28.4.1 SCG Clock Mode Transitions

The following figure shows the valid clock mode transitions supported by SCG.

Slow IRC (SIRC) boot mode is not supported on this device.

For Assessme

Freescale Confidential Proprietary - Non-Disclosure Agreement required

SCG Valid Mode Transitions

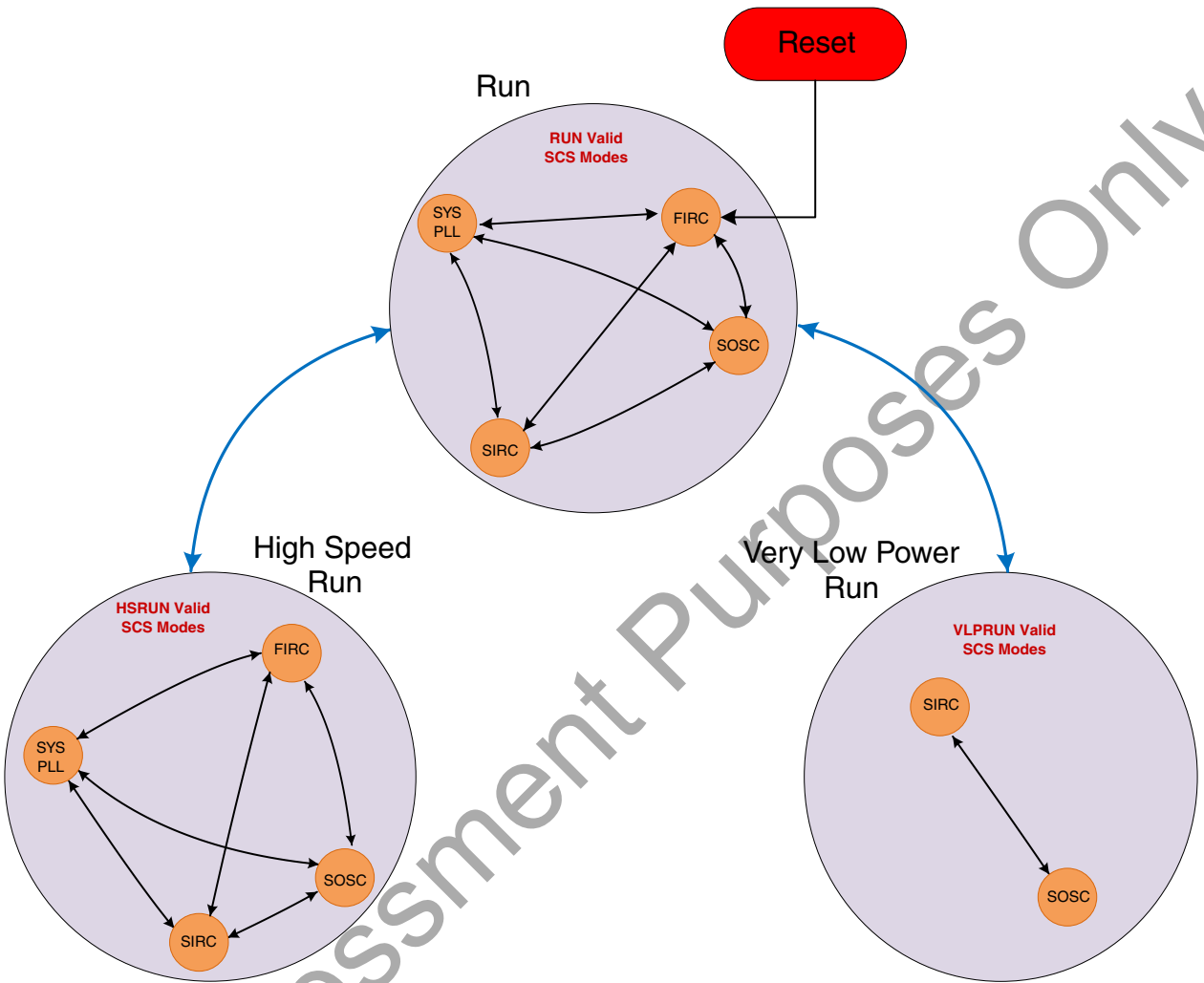


Figure 28-1. SCG Valid Mode Transition Diagram

The SCG will restrict programming into invalid clock modes and writes to SCS bits will be ignored. When a transition between run modes or a transition into wait mode occurs, the SCG completes the switch to the clock mode as defined in the SCG clock control register. When completed, the system switches to the selected run/wait mode.

The modes of operation listed in the following table are the valid modes for this implementation of the SCG.

For Assessment Purposes Only

Table 28-3. SCG modes of operation

Mode	Description
System Oscillator Clock (SOSC)	System Oscillator Clock (SOSC) mode is entered when all the following conditions occur: • RUN MODE: 0001 is written to RCCR[SCS]. VLRUN MODE: 0001 is written to VCCR[SCS]. HSRUN MODE: 0001 is written to HCCR[SCS]. • SOSCEN = 1 • SOSCVLD = 1 In SOSC mode, SCSCLKOUT and system clocks are derived from the external System Oscillator Clock (SOSC).
Slow Internal Reference Clock (SIRC)	Slow Internal Reference Clock (SIRC) mode is entered when all the following conditions occur: • RUN MODE: 0010 is written to RCCR[SCS]. VLRUN MODE: 0010 is written to VCCR[SCS] and 1 is written to SIRCCSR[SIRCLPEN]. HSRUN MODE: 0010 is written to HCCR[SCS]. • SIRCEN = 1 • SIRCVLD = 1 In SIRC mode, SCSCLKOUT and system clocks are derived from the slow internal reference clock. Two frequency ranges are available for SIRC clock as described in the SIRCCFG[RANGE] register definition. Changes to SIRC range settings will be ignored when SIRC clock is enabled.
Fast Internal Reference Clock (FIRC)	Fast Internal Reference Clock (FIRC) mode is the default clock mode of operation and is entered when all the following conditions occur: • RUN MODE: 0011 is written to RCCR[SCS]. VLRUN MODE: Invalid mode. Programming SCG into FIRC mode will be ingored. HSRUN MODE: 0011 is written to HCCR[SCS]. • FIRCEN = 1 • FIRCVLD = 1 In FIRC mode, SCSCLKOUT and system clocks are derived from the fast internal reference clock. Two frequency range settings are available for FIRC clock as described in the FIRC[RANGE] register definition. Changes to FIRC range settings will be ignored when FIRC clock is enabled.
Sys PLL (SPLL)	Sys PLL (SPLL) mode is entered when all the following conditions occur: • RUN MODE: 0110 is written to RCCR[SCS]. VLRUN MODE: Invalid mode. Programming SCG into SPLL mode will be ingored. HSRUN MODE: 0110 is written to HCCR[SCS]. • SPLLEN = 1 • SPLLVLD = 1 In SPLL mode, the SCSCLKOUT and system clocks are derived from the output of PLL which is controlled by either the System Oscillator (SOSC) clock or the Fast internal reference clock (FIRC). The selected PLL clock frequency locks to a multiplication factor, as specified by its corresponding SCG_SPLLCFG[MULT], times the selected PLL reference frequency. The PLL's programmable reference divider must be configured to produce a valid PLL reference clock. This divide value is defined by the SCG_SPLLCFG[PREDIV] bits.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 28-3. SCG modes of operation (continued)

Mode	Description
Stop	Entered whenever the MCU enters a Stop state. The power modes are chip specific. For power mode assignments, see the chapter that describes how modules are configured and SCG behaviour during Stop recovery. Entering Stop mode, all SCG clock signals are static except the following clocks which can continue to run and stayed enabled in the following cases: SIRCCLK is available in Normal Stop and VLPS mode when all the following conditions become true: • SIRCCSR[SIRCEN] = 1 • SIRCCSR[SIRCSTEN] = 1 • SIRCCSR[SIRCLPEN] = 1 in VLPS FIRCCLK is available only in Normal Stop mode when all the following conditions become true: • FIRCCSR[FIRCEN] = 1 • FIRCCSR[FIRCSTEN] = 1 SOSCLK is available in following low power stop modes (Normal Stop, VLPS) when all the below conditions are true. • SOSCCSR[SOSCEN] = 1 • SOSCCSR[SOSCSTEN] = 1 • SOSCCSR[SOSCLPEN] = 1 (required only for Low Power Stop modes (VLPS) SPLLCLK is available in Normal Stop mode when all the following conditions are true: • SPLLCSR[SPLLEN] = 1 • SPLLCSR[SPLLSTEN] = 1 • SPLLSTEN control bit has no affect in VLPS Power mode.

For Assessm

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 29
Peripheral Clock Controller (PCC)

29.1 Introduction

The Peripheral Clock Control block (PCC) provides peripheral clock control and configuration registers. Each peripheral has its own clock control and configuration register. With the PCC, all the clock gates are centralized in one block, with one PCC register for each peripheral on the MCU. For the peripheral's operational clock, it must be enabled in the PCC register before the peripheral can be used. A peripheral may also require an asynchronous clock so the PCC also provides three banks of clock inputs for a peripheral with up to seven clocks in each bank, which are software selectable. Any peripheral can have a choice of input clock from the bank of clocks for its asynchronous clock by setting the Peripheral Clock Select (PCS) bits. Optionally, any peripheral can also have a clock divider (either four bit or 24-bit), for its asynchronous clock, seleted by setting the Peripheral Clock Divider (PCD) bits along with the Fractional clock divide bit (FRAC).

29.2 Features

The PCC module enables software to configure the following clocking options for each peripheral:

- Clock gating
- Clock source selection
- Clock divide values

For Assessment Purposes Only

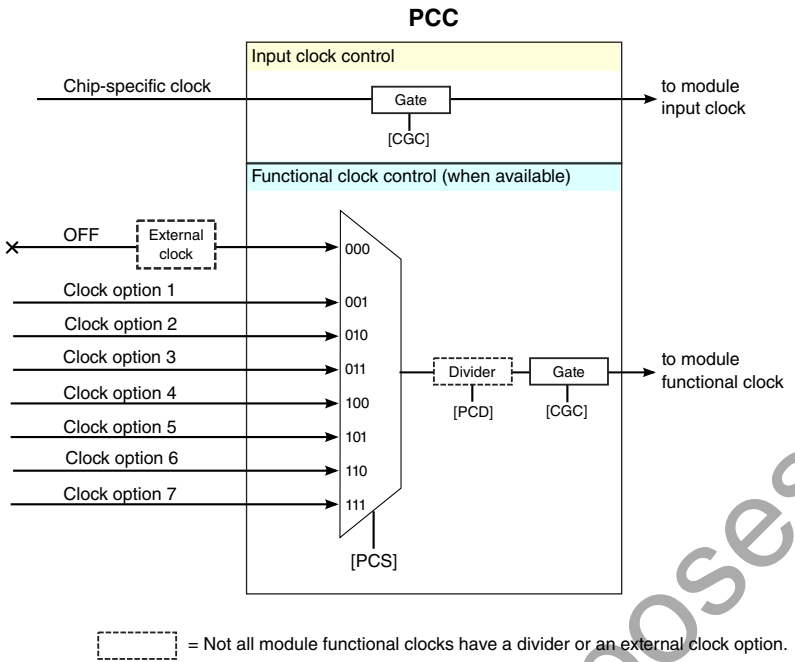


Figure 29-1. PCC Block Diagram

29.3 Functional description

The Peripheral Clock Control (PCC) module provides clock gating and clock source selection to each peripheral.

Each peripheral has its own unique PCCn register that provides clock gating for the peripheral's interface clock and where applicable, its functional clock.

See each peripheral's PCCn register for details.

29.4 Memory map and register definition

Each peripheral has its own dedicated PCC Register, which controls the clock gating, clock source and dividers for that specific peripheral.

29.5 PCC Register Descriptions

For Assessment Purposes Only

29.5.1 PCC Memory Map

Offset	Register	Width (In bits)	Access	Reset value
20h	PCC DMA (PCC_DMA)	32	RW	C0000000h
34h	PCC MPU (PCC_MPU)	32	RW	C0000000h
80h	PCC Flash (PCC_Flash)	32	RW	C0000000h
84h	PCC DMA_channel_mutiplexer (PCC_DMA_channel_mutiplexer)	32	RW	80000000h
90h	PCC FlexCAN_0 (PCC_FlexCAN_0)	32	RW	80000000h
94h	PCC FlexCAN_1 (PCC_FlexCAN_1)	32	RW	80000000h
98h	PCC FTM_3 (PCC_FTM_3)	32	RW	80000000h
9Ch	PCC ADC_1 (PCC_ADC_1)	32	RW	C0000000h
ACh	PCC FlexCAN_2 (PCC_FlexCAN_2)	32	RW	80000000h
B0h	PCC LPSPI_0 (PCC_LPSPI_0)	32	RW	80000000h
B4h	PCC LPSPI_1 (PCC_LPSPI_1)	32	RW	80000000h
B8h	PCC LPSPI_2 (PCC_LPSPI_2)	32	RW	80000000h
C4h	PCC PDB1 (PCC_PDB1)	32	RW	80000000h
C8h	PCC CRC (PCC_CRC)	32	RW	80000000h
D8h	PCC PDB0 (PCC_PDB0)	32	RW	80000000h
DCh	PCC LPIT0 (PCC_LPIT0)	32	RW	80000000h
E0h	PCC FTM0 (PCC_FTM0)	32	RW	80000000h
E4h	PCC FTM1 (PCC_FTM1)	32	RW	80000000h
E8h	PCC FTM2 (PCC_FTM2)	32	RW	80000000h
ECh	PCC ADC0 (PCC_ADC0)	32	RW	C0000000h
F4h	PCC RTC (PCC_RTC)	32	RW	80000000h
100h	PCC LPTMR0 (PCC_LPTMR0)	32	RW	80000000h
118h	PCC (PCC_)	32	RW	80000000h
124h	PCC PORTA (PCC_PORTA)	32	RW	80000000h
128h	PCC PORTB (PCC_PORTB)	32	RW	80000000h
12Ch	PCC PORTC (PCC_PORTC)	32	RW	80000000h
130h	PCC PORTD (PCC_PORTD)	32	RW	80000000h
134h	PCC PORTE (PCC_PORTE)	32	RW	80000000h
168h	PCC FlexIO (PCC_FlexIO)	32	RW	80000000h
184h	PCC EWM (PCC_EWM)	32	RW	C0000000h
198h	PCC LPI2C0 (PCC_LPI2C0)	32	RW	80000000h
1A8h	PCC LPUART0 (PCC_LPUART0)	32	RW	80000000h
1ACh	PCC LPUART1 (PCC_LPUART1)	32	RW	80000000h
1B0h	PCC LPUART2 (PCC_LPUART2)	32	RW	80000000h
1CCh	PCC CMP0 (PCC_CMP0)	32	RW	80000000h

29.5.2 PCC DMA (PCC_DMA)

29.5.2.1 Address

Register	Offset
PCC_DMA	20h

29.5.2.2 Function

PCC Register

29.5.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.2.4 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29	Clock Gate Status

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
INUSE	This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.3 PCC MPU (PCC_MPU)

29.5.3.1 Address

Register	Offset
PCC_MPU	34h

PCC Register

29.5.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.3.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.4 PCC Flash (PCC_Flash)

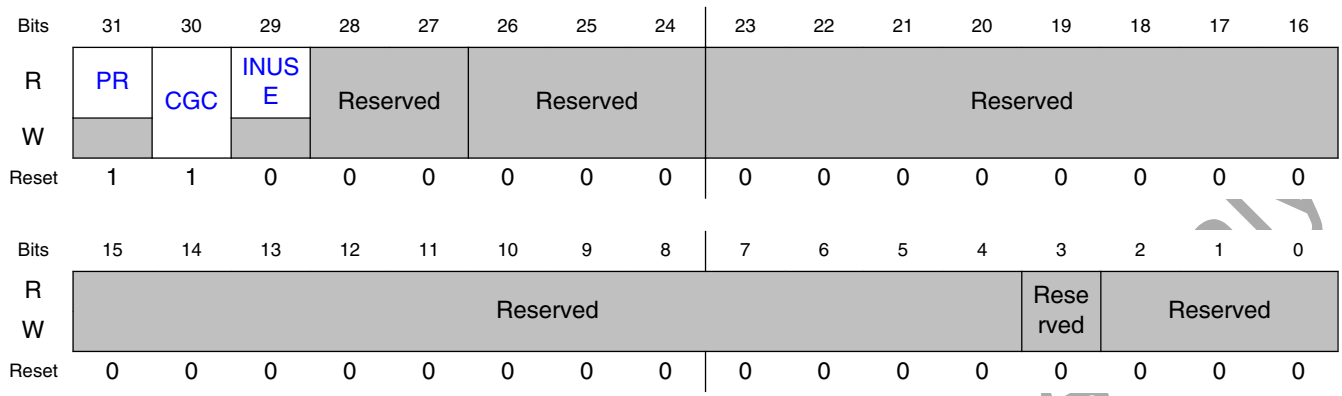
29.5.4.1 Address

Register	Offset
PCC_Flash	80h

PCC Register

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.4.2 Diagram



29.5.4.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.5 PCC DMA_channel_mutiplexer (PCC_DMA_channel_mutiplexer)

29.5.5.1 Address

Register	Offset
PCC_DMA_channel_mutiplexer	84h

PCC Register

29.5.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.5.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29	Clock Gate Status

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
INUSE	This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.6 PCC FlexCAN_0 (PCC_FlexCAN_0)

29.5.6.1 Address

Register	Offset
PCC_FlexCAN_0	90h

PCC Register

29.5.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Assessment Purpos

29.5.6.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.7 PCC FlexCAN_1 (PCC_FlexCAN_1)

29.5.7.1 Address

Register	Offset
PCC_FlexCAN_1	94h

PCC Register

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.7.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.8 PCC FTM_3 (PCC_FTM_3)

29.5.8.1 Address

Register	Offset
PCC_FTM_3	98h

PCC Register

29.5.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		PCS			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.8.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. An external clock can be enabled for this peripheral. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.9 PCC ADC_1 (PCC_ADC_1)

29.5.9.1 Address

Register	Offset
PCC_ADC_1	9Ch

PCC Register

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		PCS			Reserved							
W																
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.9.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.10 PCC FlexCAN_2 (PCC_FlexCAN_2)

29.5.10.1 Address

Register	Offset
PCC_FlexCAN_2	ACh

PCC Register

29.5.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.10.3 Fields

Field	Function
31	Enable
PR	This bit shows whether the peripheral is present on this device.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.11 PCC LPSPI_0 (PCC_LPSPI_0)

29.5.11.1 Address

Register	Offset
PCC_LPSPI_0	B0h

PCC Register

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		PCS			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.11.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.12 PCC LPSPI_1 (PCC_LPSPI_1)

29.5.12.1 Address

Register	Offset
PCC_LPSPI_1	B4h

PCC Register

29.5.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		PCS			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.12.3 Fields

Field	Function
31	Enable
PR	This bit shows whether the peripheral is present on this device.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

For Asse

29.5.13 PCC LPSPI_2 (PCC_LPSPI_2)

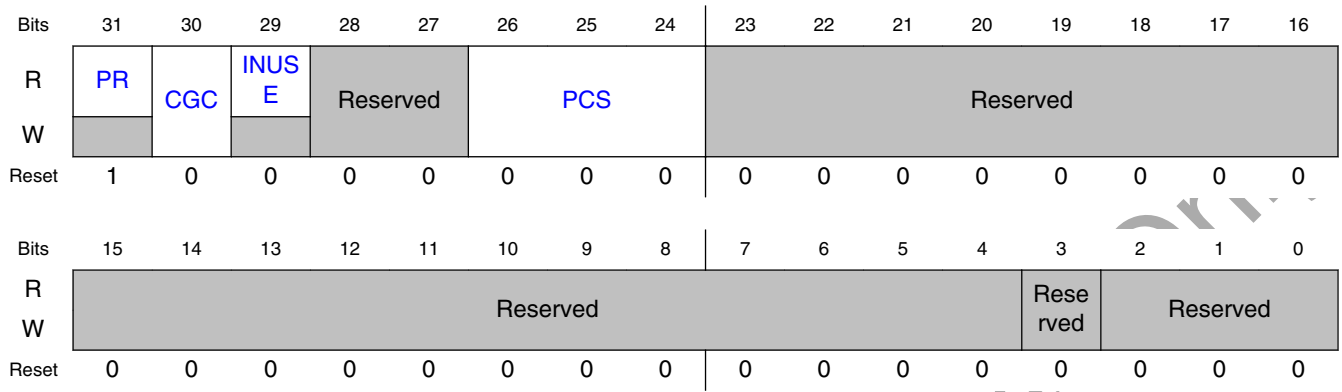
29.5.13.1 Address

Register	Offset
PCC_LPSPI_2	B8h

PCC Register

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.13.2 Diagram



29.5.13.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.14 PCC PDB1 (PCC_PDB1)

29.5.14.1 Address

Register	Offset
PCC_PDB1	C4h

PCC Register

29.5.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.14.3 Fields

Field	Function
31	Enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
PR	This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.15 PCC CRC (PCC_CRC)

29.5.15.1 Address

Register	Offset
PCC_CRC	C8h

PCC Register

For Assessment

29.5.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.15.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.16 PCC PDB0 (PCC_PDB0)

29.5.16.1 Address

Register	Offset
PCC_PDB0	D8h

PCC Register

29.5.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.16.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.17 PCC LPIT0 (PCC_LPIT0)

29.5.17.1 Address

Register	Offset
PCC_LPIT0	DCh

PCC Register

29.5.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		PCS			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Freescale Confidential Proprietary Non-Disclosure Agreement required

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.17.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

For

29.5.18 PCC FTM0 (PCC_FTM0)

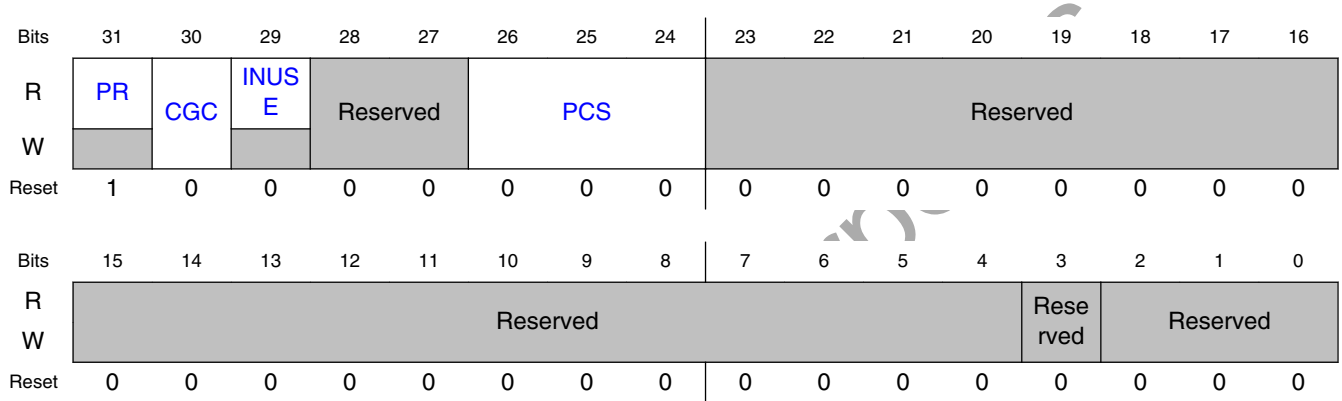
Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.18.1 Address

Register	Offset
PCC_FTM0	E0h

PCC Register

29.5.18.2 Diagram



29.5.18.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24	Peripheral Clock Source Select

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
PCS	This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. An external clock can be enabled for this peripheral. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.19 PCC FTM1 (PCC_FTM1)

29.5.19.1 Address

Register	Offset
PCC_FTM1	E4h

PCC Register

29.5.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		PCS			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.19.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. An external clock can be enabled for this peripheral. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.20 PCC FTM2 (PCC_FTM2)

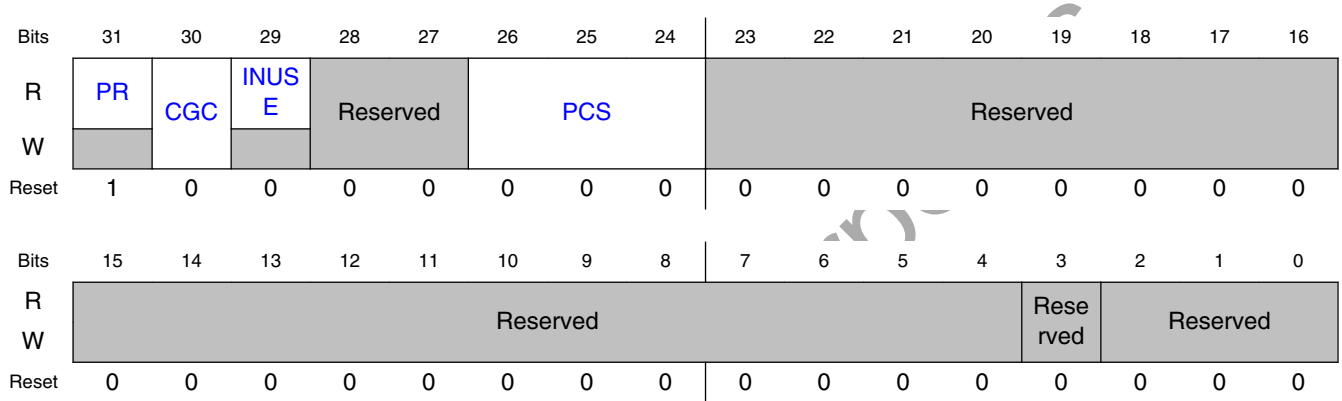
Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.20.1 Address

Register	Offset
PCC_FTM2	E8h

PCC Register

29.5.20.2 Diagram



29.5.20.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24	Peripheral Clock Source Select

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
PCS	This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. An external clock can be enabled for this peripheral. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.21 PCC ADC0 (PCC_ADC0)

29.5.21.1 Address

Register	Offset
PCC_ADC0	ECh

PCC Register

29.5.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		PCS			Reserved							
W																
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.21.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.22 PCC RTC (PCC_RTC)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.22.1 Address

Register	Offset
PCC_RTC	F4h

PCC Register

29.5.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.22.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
—	
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

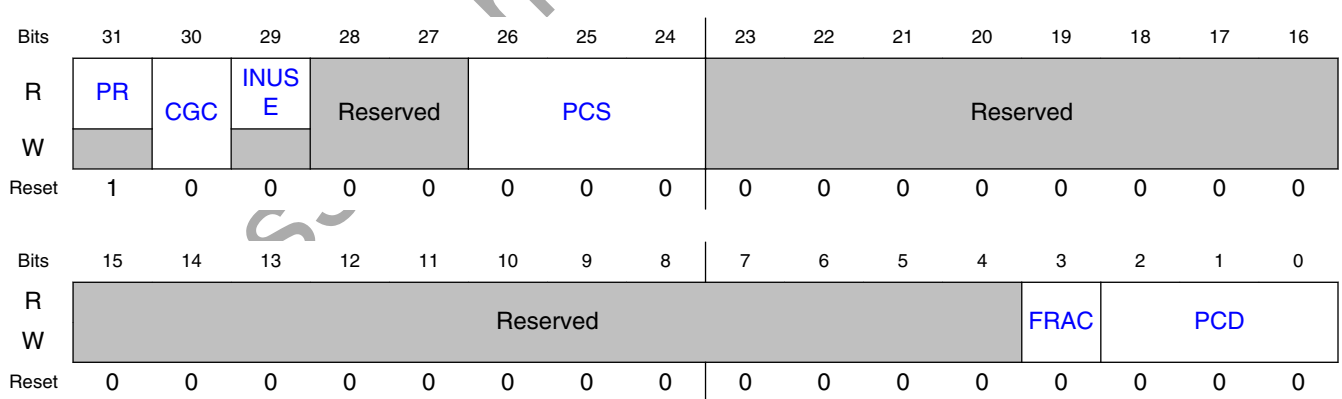
29.5.23 PCC LPTMR0 (PCC_LPTMR0)

29.5.23.1 Address

Register	Offset
PCC_LPTMR0	100h

PCC Register

29.5.23.2 Diagram



29.5.23.3 Fields

Field	Function
31	Enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
PR	This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 FRAC	Peripheral Clock Divider Fraction This read/write bit field sets the fraction multiply value for the fractional clock divider used as a clock source. Divider output clock = Divider input clock x [(FRAC+1)/(DIV+1)]. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. The FRAC bit should not be set if the PCD bits are set to all zeroes, or the output clock will be disabled. 0b - Fractional value is 0. 1b - Fractional value is 1.
2-0 PCD	Peripheral Clock Divider Select This read/write bit field is used for peripherals that require a clock divider. Divider output clock = Divider input clock x [(FRAC+1)/(DIV+1)]. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Divide by 1 (pass-through, no clock divide). 001b - Divide by 2. 010b - Divide by 3. 011b - Divide by 4. 100b - Divide by 5. 101b - Divide by 6. 110b - Divide by 7.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	111b - Divide by 8.

29.5.24 PCC (PCC_)

29.5.24.1 Address

Register	Offset
PCC_	118h

PCC Register

29.5.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.24.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.25 PCC PORTA (PCC_PORTA)

29.5.25.1 Address

Register	Offset
PCC_PORTA	124h

PCC Register

For Assessment Purp

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.25.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.26 PCC PORTB (PCC_PORTB)

29.5.26.1 Address

Register	Offset
PCC_PORTB	128h

PCC Register

29.5.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.26.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.27 PCC PORTC (PCC_PORTC)

29.5.27.1 Address

Register	Offset
PCC_PORTC	12Ch

PCC Register

29.5.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Freescale Confidential Proprietary Internal Use Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.27.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.28 PCC PORTD (PCC_PORTD)

29.5.28.1 Address

Register	Offset
PCC_PORTD	130h

PCC Register

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.28.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.29 PCC PORTE (PCC_PORTE)

29.5.29.1 Address

Register	Offset
PCC_PORTE	134h

PCC Register

29.5.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.29.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.30 PCC FlexIO (PCC_FlexIO)

29.5.30.1 Address

Register	Offset
PCC_FlexIO	168h

PCC Register

29.5.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		PCS			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Freescale Confidential Proprietary Non-Disclosure Agreement required

29.5.30.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

For

29.5.31 PCC EWM (PCC_EWM)

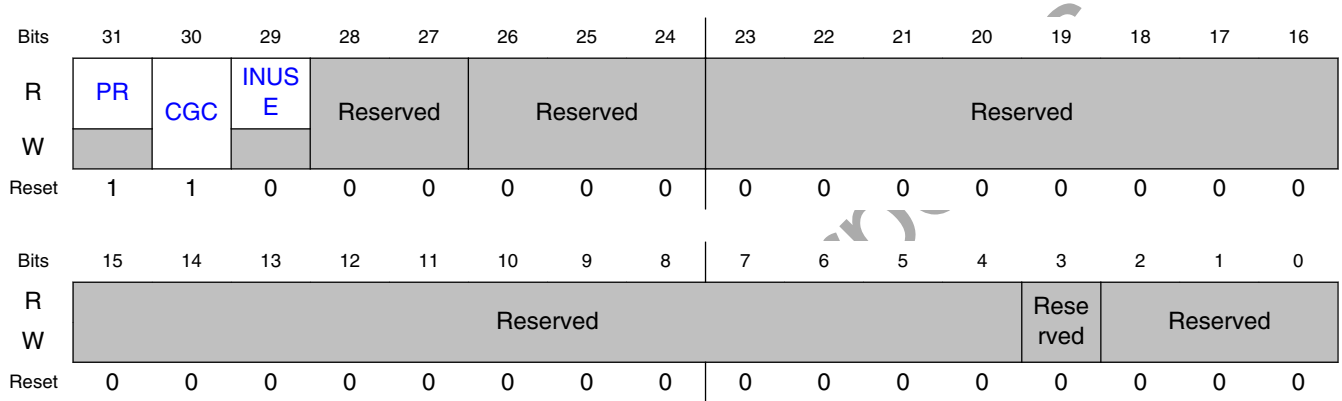
Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.31.1 Address

Register	Offset
PCC_EWM	184h

PCC Register

29.5.31.2 Diagram



29.5.31.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24	This read-only bit field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
—	
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.32 PCC LPI2C0 (PCC_LPI2C0)

29.5.32.1 Address

Register	Offset
PCC_LPI2C0	198h

PCC Register

29.5.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		PCS			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.32.3 Fields

Field	Function
31	Enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
PR	This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

For Ass

29.5.33 PCC LPUART0 (PCC_LPUART0)

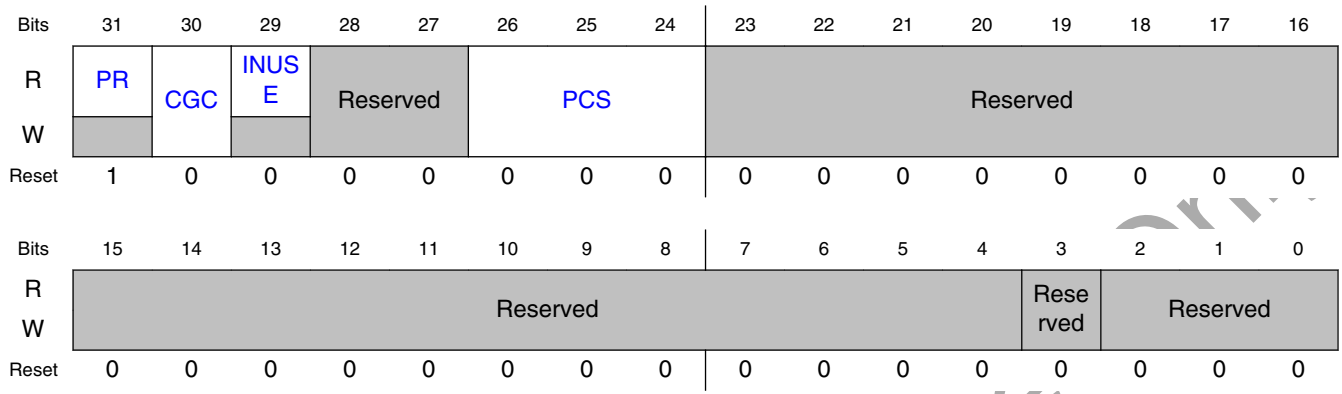
29.5.33.1 Address

Register	Offset
PCC_LPUART0	1A8h

PCC Register

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.33.2 Diagram



29.5.33.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.34 PCC LPUART1 (PCC_LPUART1)

29.5.34.1 Address

Register	Offset
PCC_LPUART1	1ACh

PCC Register

29.5.34.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		PCS			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.34.3 Fields

Field	Function
31	Enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
PR	This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

For Ass

29.5.35 PCC LPUART2 (PCC_LPUART2)

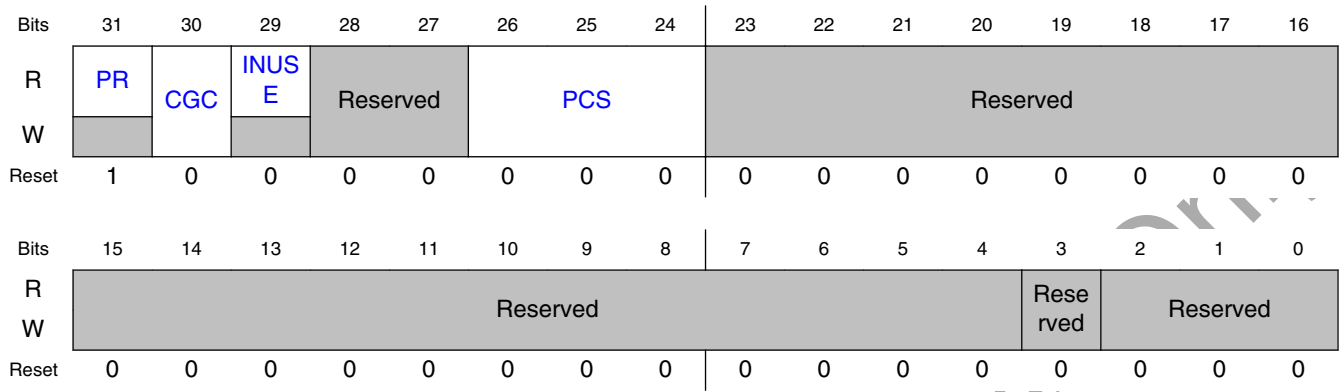
29.5.35.1 Address

Register	Offset
PCC_LPUART2	1B0h

PCC Register

Freescale Confidential Proprietary - Non-Disclosure Agreement required

29.5.35.2 Diagram



29.5.35.3 Fields

Field	Function
31 PR	Enable This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can only be written when the CGC bit is 0 (clock disabled). Likewise, if the INUSE flag is set, this field is locked. 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	111b - Clock option 7
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

29.5.36 PCC CMP0 (PCC_CMP0)

29.5.36.1 Address

Register	Offset
PCC_CMP0	1CCh

PCC Register

29.5.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PR	CGC	INUSE	Reserved		Reserved			Reserved							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

29.5.36.3 Fields

Field	Function
31	Enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
PR	This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. Software cannot modify the existing clocking configuration.
29 INUSE	Clock Gate Status This read-only bit shows that this peripheral is being used. 0b - Peripheral is not being used. 1b - Peripheral is being used. Software cannot modify the existing clocking configuration.
28-27 —	This read-only bit field is reserved and always has the value 0.
26-24 —	This read-only bit field is reserved and always has the value 0.
23-4 —	This read-only bit field is reserved and always has the value 0.
3 —	This read-only bit field is reserved and always has the value 0.
2-0 —	This read-only bit field is reserved and always has the value 0.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 30
Local Memory Controller (LMEM)

30.1 Introduction

The Local Memory Controller provides the ARM®Cortex®-M4 processor with tightly-coupled processor-local memories and bus paths to all slave memory spaces.

30.1.1 Block Diagram

The Cortex-M4 processor has a modified 32-bit Harvard bus architecture. Using a 32-bit address space, low-order addresses (0x0000_0000 through 0x1FFF_FFFF) use the Processor Code (PC) bus, and high-order addresses (0x2000_0000 through 0xFFFF_FFFF) use the Processor System (PS) bus. As the bus names imply, normal operation has code accesses on the PC bus and data accesses on the PS bus.

This device has been augmented with tightly-coupled memories for the PC and PS buses. The memories include RAMs and caches. These local memories provide zero wait state access to RAM and cacheable address spaces.

The local memory controller includes three memory controllers and their attached memories:

- SRAM lower (SRAM_L) controller via the PC bus
- SRAM upper (SRAM_U) controller via the PS bus
- Cache memory controller via the PC bus

The local memory controller has the following AMBA_ AHB buses:

- Two inputs are for the CM4's modified Harvard busses – the Processor Code (PC) and the Processor System (PS) buses.

For Assessment Purposes Only

- One input is for the "backdoor" port used by all other bus masters to access the SRAM controller space.
- Two output ports are the CCM (Core Code Master) bus used for PC accesses that do not hit the PC cache or SRAM_L or are non-cacheable and the CSM (Core System Master) bus used for PS references that do not hit the SRAM_U.

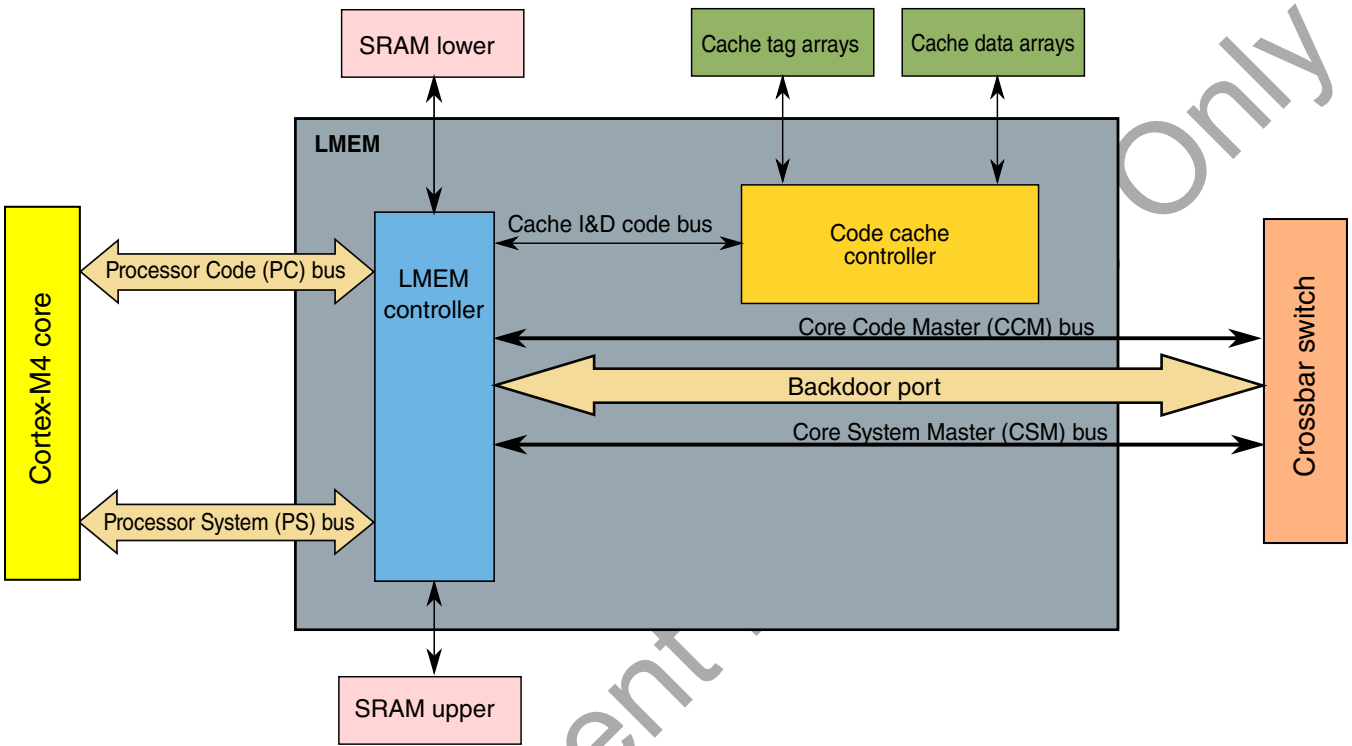


Figure 30-1. Local memory controller block diagram

NOTE

The SRAM and cache controllers reside within the LMEM, but the single-port synchronous RAM arrays used by these controllers are external.

The LMEM contains address decode logic for the PC and PS buses. This logic routes the core's accesses to the various system resources. The address spaces are device-specific. See the chip-specific LMEM information for the address space decode details.

30.1.2 Cache features

A cache is a block of high-speed memory locations containing address information (commonly known as a tag) and the associated data. The purpose is to decrease the average time of a memory access. Caches operate on two principles of locality:

For Assessment Only

- Spatial locality — An access to one location is likely to be followed by accesses from adjacent locations (for example, sequential instruction execution or usage of a data structure).
- Temporal locality — An access to an area of memory is likely to be repeated within a short time period (for example, execution of a code loop).

To minimize the quantity of control information stored, the spatial locality property is used to group several locations together under the same tag. This logical block is commonly known as a cache line.

When data is loaded into a cache, access times for subsequent loads and stores are reduced, resulting in overall performance benefits. An access to information already in a cache is known as a cache hit, and other accesses are called cache misses.

Normally, caches are self-managing, with the updates occurring automatically. Whenever the processor wants to access a cacheable location, the cache is checked. If the access is a cache hit, the access occurs immediately. Otherwise, a location is allocated and the cache line is loaded from memory. Different cache topologies and access policies are possible. However, they must comply with the memory coherency model of the underlying architecture.

Caches introduce a number of potential problems, mainly because of:

- memory accesses occurring at times other than when the programmer would normally expect them,
- the existence of multiple physical locations where a data item can be held.

The local memory controller supports the following modes of operation:

1. Write-through — access to address spaces with this cache mode are cacheable.
   - A write-through read miss on the input bus causes a line read on the output bus of a 16-byte-aligned memory address containing the desired address. This miss data is loaded into the cache and is marked as valid and not modified.
   - A write-through read hit to a valid cache location returns data from the cache with no output bus access.
   - A write-through write miss bypasses the cache and writes to the output bus (no allocate on write miss policy for write-through mode spaces).
   - A write-through write hit updates the cache hit data and writes to the output bus.
   - The caches are processor-local and do not support hardware cache coherency. If the processor has accessed write-through regions and an external bus master (such as DMA) then needs update these regions, software must first perform explicit cache clears to any needed cache memory range to ensure all modified cache lines update their associated memories before being modified by external masters and subsequent processor accesses will get the updated memory.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

2. Non-cacheable — access to address spaces with this cache mode are not cacheable. These accesses bypass the cache and access the output bus.

30.2 Memory Map/Register Definition

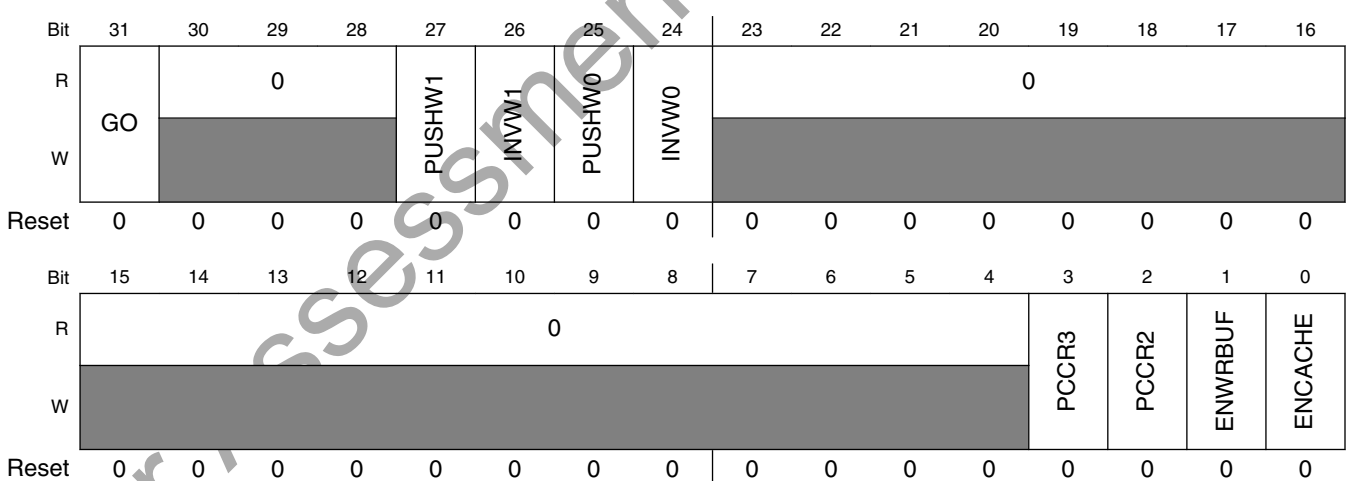
The cache programmer's model provides a variety of registers for configuring and controlling the cache, as well as indirect access paths to all cache tag and data storage.

LMEM memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Cache control register (LMEM_PCCCR)	32	R/W	0000_0000h	30.2.1/568
4	Cache line control register (LMEM_PCCLCR)	32	R/W	0000_0000h	30.2.2/569
8	Cache search address register (LMEM_PCCSAR)	32	R/W	0000_0000h	30.2.3/572
C	Cache read/write value register (LMEM_PCCCVR)	32	R/W	0000_0000h	30.2.4/573
20	Cache regions mode register (LMEM_PCCRMR)	32	R/W	AA0F_A000h	30.2.5/573

30.2.1 Cache control register (LMEM_PCCCR)

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



LMEM_PCCCR field descriptions

Field	Description
31 GO	Initiate Cache Command Setting this bit initiates the cache command indicated by bits 27-24. Reading this bit indicates if a command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

LMEM_PCCCR field descriptions (continued)

Field	Description
	0 Write: no effect. Read: no cache command active. 1 Write: initiate command indicated by bits 27-24. Read: cache command active.
30–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 PUSHW1	Push Way 1 0 No operation 1 When setting the GO bit, push all modified lines in way 1
26 INVW1	Invalidate Way 1 **NOTE:** If the PUSHW1 and INVW1 bits are set, then after setting the GO bit, push all modified lines in way 1 and invalidate all lines in way 1 (clear way 1). 0 No operation 1 When setting the GO bit, invalidate all lines in way 1
25 PUSHW0	Push Way 0 0 No operation 1 When setting the GO bit, push all modified lines in way 0
24 INVW0	Invalidate Way 0 **NOTE:** If the PUSHW0 and INVW0 bits are set, then after setting the GO bit, push all modified lines in way 0 and invalidate all lines in way 0 (clear way 0). 0 No operation 1 When setting the GO bit, invalidate all lines in way 0.
23–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 PCCR3	Forces no allocation on cache misses (must also have PCCR2 asserted)
2 PCCR2	Forces all cacheable spaces to write through
1 ENWRBUF	Enable Write Buffer 0 Write buffer disabled 1 Write buffer enabled
0 ENCACHE	Cache enable 0 Cache disabled 1 Cache enabled

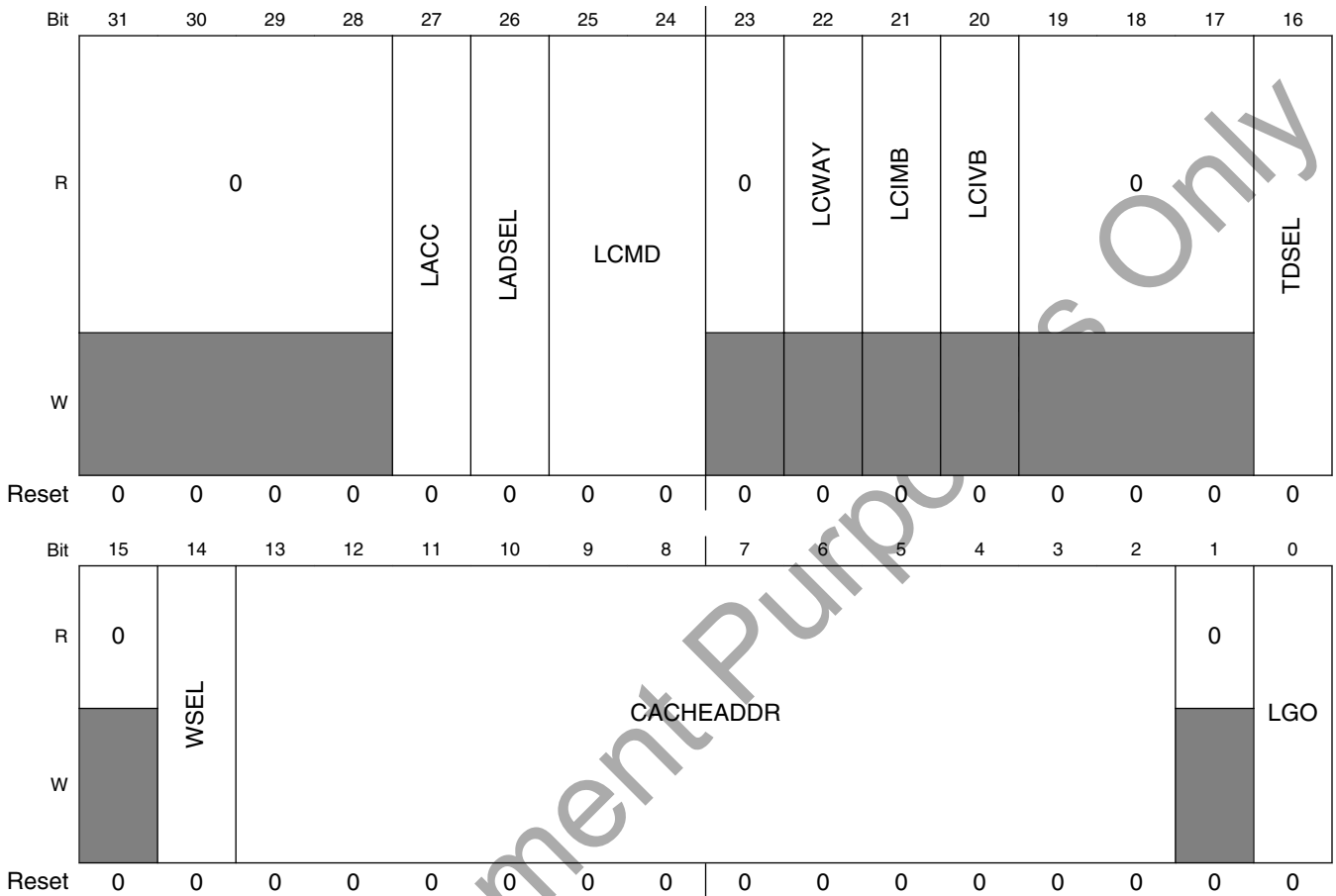
30.2.2 Cache line control register (LMEM_PCCLCR)

This register defines specific line-sized cache operations to be performed using a specific cache line address or a physical address.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

If a physical address is specified, both ways of the cache are searched, and the command is only performed on the way which hits.

Address: 0h base + 4h offset = 4h



ment Purposes Only

LMEM_PCCLCR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 LACC	Line access type 0 Read 1 Write
26 LADSEL	Line Address Select When using the cache address, the way must also be specified in CLCR[WSEL]. When using the physical address, both ways are searched and the command is performed only if a hit. 0 Cache address 1 Physical address
25–24 LCMD	Line Command 00 Search and read or write

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

LMEM_PCCLCR field descriptions (continued)

Field	Description
	01 Invalidate 10 Push 11 Clear
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 LCWAY	Line Command Way Indicates the way used by the line command. Only applies if valid bit LCIVB = 1.
21 LCIMB	Line Command Initial Modified Bit If command used cache address and way, then this bit shows the initial state of the modified bit If command used physical address and a hit, then this bit shows the initial state of the modified bit. If a miss, this bit reads zero.
20 LCIVB	Line Command Initial Valid Bit If command used cache address and way, then this bit shows the initial state of the valid bit If command used physical address and a hit, then this bit shows the initial state of the valid bit. If a miss, this bit reads zero.
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 TDSEL	Tag/Data Select Selects tag or data for search and read or write commands. 0 Data 1 Tag
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 WSEL	Way select Selects the way for line commands. 0 Way 0 1 Way 1
13–2 CACHEADDR	Cache address CLCR[10:4] bits are used to access the tag arrays CLCR[10:2] bits are used to access the data arrays
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CSAR[LGO]

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

LMEM_PCCLCR field descriptions (continued)

Field	Description
	0 Write: no effect. Read: no line command active. 1 Write: initiate line command indicated by bits 27-24. Read: line command active.

30.2.3 Cache search address register (LMEM_PCCSAR)

The CSAR register is used to define the explicit cache address or the physical address for line-sized commands specified in the CLCR[LADSEL] bit.

Address: 0h base + 8h offset = 8h

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

LMEM_PCCSAR field descriptions

Field	Description
31–2 PHYADDR	Physical Address PHYADDR represents bits [31:2] of the system address. CSAR[31:11] bits are used for tag compare CSAR[10:4] bits are used to access the tag arrays CSAR[10:2] bits are used to access the data arrays
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CLCR[LGO] 0 Write: no effect. Read: no line command active. 1 Write: initiate line command indicated by bits CLCR[27:24]. Read: line command active.

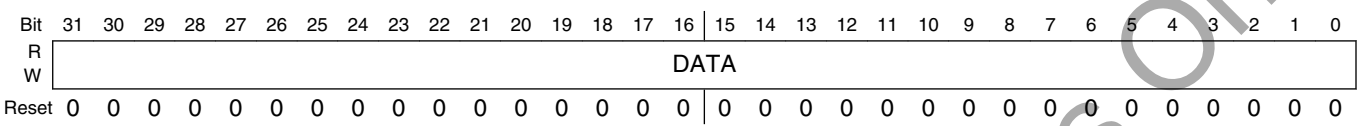
Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

30.2.4 Cache read/write value register (LMEM_PCCCVR)

The CCVR register is used to source write data or return read data for the commands specified in the CLCR register.

Address: 0h base + Ch offset = Ch



LMEM_PCCCVR field descriptions

Field	Description
DATA	Cache read/write Data For tag search, read or write: • CCVR[31:11] bits are used for tag array R/W value • CCVR[10:4] bits are used for tag set address on reads; unused on writes • CCVR[3:2] bits are reserved • CCVR[1] tag modify bit • CCVR[0] tag valid bit For data search, read or write: • CCVR[31:0] bits are used for data array R/W value

30.2.5 Cache regions mode register (LMEM_PCCRMR)

The CRMR register allows you to demote the cache mode of various subregions within the device's memory map. Demoting the cache mode reduces the cache function applied to a memory region from write-back to write-through to non-cacheable. After a region is demoted, its cache mode can only be raised by a reset, which returns it to its default state.

To maintain cache coherency, changes to the cache mode should be completed while the address space being changed is not being accessed or the cache is disabled. Before a cache mode change, complete a cache clear all command to push and invalidate any cache entries that may have changed.

NOTE

The address/module assignment of the 16 subregions is device-specific. See the chip-specific LMEM information for these details. Some of the regions may not be used (non-cacheable), and some regions may not be capable of write-back.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 20h offset = 20h

Bit	31	30	29	28	27	26	25	24	23	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	R0				R1				R2				R3				R4				R5				R6				R7			
Reset	1	0	1	0	1	0	1	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	R8		R9		R10		R11		R12		R13		R14		R15	
Reset	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PCCRMR field descriptions

Field	Description
31–30 R0	Region 0 mode Controls the cache mode for region 0 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
29–28 R1	Region 1 mode Controls the cache mode for region 1 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
27–26 R2	Region 2 mode Controls the cache mode for region 2 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
25–24 R3	Region 3 mode Controls the cache mode for region 3 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
23–22 R4	Region 4 mode Controls the cache mode for region 4 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
21–20 R5	Region 5 mode Controls the cache mode for region 5 00 Non-cacheable 01 Non-cacheable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

LMEM_PCCRMR field descriptions (continued)

Field	Description
	10 Write-through 11 Write-back
19–18 R6	Region 6 mode Controls the cache mode for region 6 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
17–16 R7	Region 7 mode Controls the cache mode for region 7 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
15–14 R8	Region 8 mode Controls the cache mode for region 8 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
13–12 R9	Region 9 mode Controls the cache mode for region 9 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
11–10 R10	Region 10 mode Controls the cache mode for region 10 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
9–8 R11	Region 11 mode Controls the cache mode for region 11 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

LMEM_PCCRMR field descriptions (continued)

Field	Description
7–6 R12	Region 12 mode Controls the cache mode for region 12 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
5–4 R13	Region 13 mode Controls the cache mode for region 13 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
3–2 R14	Region 14 mode Controls the cache mode for region 14 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back
R15	Region 15 mode Controls the cache mode for region 15 00 Non-cacheable 01 Non-cacheable 10 Write-through 11 Write-back

For Assess

30.3 Functional Description

30.3.1 LMEM Function

The Local Memory Controller receives the following requests:

- Core master bus requests on the Processor Code (PC) bus,
- Core master bus requests on the Processor System (PS) bus, and
- SRAM controller requests from all other bus masters on the backdoor port.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The Local Memory Controller address decode logic routes these accesses and also provides any crossbar switch slave target logic. Finally, the Local Memory controller provides the needed MPU connections for checking all SRAM controller and cacheable accesses.

The programming model for the Code Cache is accessed via the core's Private Peripheral Bus (PPB).

30.3.1.1 Processor Code accesses

Processor Code accesses are routed to the SRAM_L if they are mapped to that space. All other PC accesses are routed to the Code Cache Memory Controller. This controller then processes the cacheable accesses as needed, while bypassing the non-cacheable, cache write-through, cache miss, and cache maintenance accesses to the CCM bus and the crossbar switch using the Master0 port.

30.3.1.2 Processor System accesses

Processor System accesses are routed to the SRAM_U if they are mapped to that space. All other PS accesses are routed to the CCM bus and the crossbar switch using the Master1 port.

30.3.1.3 Backdoor port accesses

All LMEM backdoor port accesses are for the SRAM controller. These accesses go to the SRAM_L or the SRAM_U depending on their specific address.

30.3.2 SRAM Function

30.3.2.1 SRAM Configuration

The figure below shows how the SRAM controller is configured.

For Assessment Purposes Only

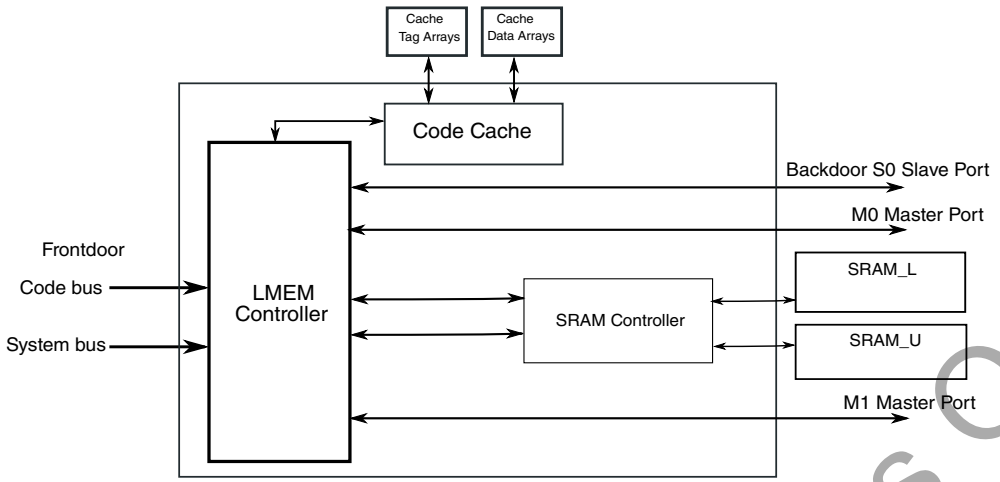


Figure 30-2. SRAM configuration

30.3.2.2 SRAM Arrays

The on-chip SRAM is split into two logical arrays, SRAM_L and SRAM_U.

Valid address ranges for SRAM_L and SRAM_U are then defined as:

- SRAM_L = (0x20000_0000 - SRAML_SIZE) - 0x1fff_ffff
- SRAM_U = 0x2000_0000 - (0x2000_0000 + SRAMU_SIZE)

SRAML_SIZE and SRAMU_SIZE do not have to be equal.

For example, if SRAM_L size is 32 KBytes and SRAM_U size is 64 KBytes. Valid address ranges for SRAM_L and SRAM_U are then defined as:

- SRAM_L = (0x20000_0000 - 32 KBytes) - 0x1fff_ffff
- SRAM_U = 0x2000_0000 - (0x2000_0000 + 64 KBytes)

30.3.2.3 SRAM accesses

The SRAM is split into two logical arrays that are 32-bits wide:

- SRAM_L — Accessible by the code bus of the Cortex-M4 core and by the backdoor port.
- SRAM_U — Accessible by the system bus of the Cortex-M4 core and by the backdoor port.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The backdoor port makes the SRAM accessible to the non-core bus masters (such as DMA).

Figure 30-3 illustrates the SRAM accesses within the device.

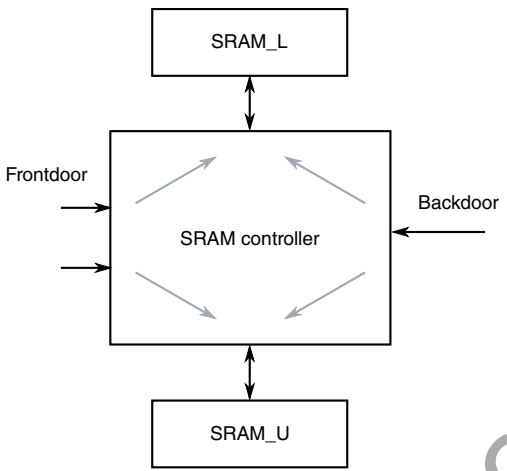


Figure 30-3. SRAM access diagram

The following simultaneous accesses can be made to different logical halves of the SRAM:

- Core code and core system
- Core code and non-core master
- Core system and non-core master

NOTE

Two non-core masters cannot access SRAM simultaneously. The required arbitration and serialization is provided by the crossbar switch. The SRAM_{L,U} arbitration is controlled by the SRAM controller based on the configuration bits in the MCM module.

NOTE

Burst-access cannot occur across the 0x2000_0000 boundary that separates the two SRAM arrays. The two arrays should be treated as separate memory ranges for burst accesses.

30.3.3 Cache Function

The cache on this device is structured as follows. The cache has a 2-way set-associative cache structure with a total size of 4 KB. The cache has 32-bit address and data paths and a 16-byte line size. The cache tags and data storage use single-port, synchronous RAMs.

For Assessment Purposes Only

For this 4 KB cache, each cache TAG function uses two 128 x 23-bit RAM arrays and the cache DATA function uses two 512 x 32-bit RAM arrays. The cache TAG entries store 21 bits of upper address as well as a modified and valid bit per cache line. The cache DATA entries store four bytes of code or data.

All normal cache accesses use physical addresses. This leads to the following cache address use:

CACHE - 4 KB size = (128 sets) x (16-byte lines) x (2-way set-associative)

TAG:

- address[31:11] used in tag for compare (hit) logic
- address[10:4] used to select 1 of 128 sets
- address[3:0] not used

DATA

- address[31:11] not used
- address[10:4] used to select one of 128 sets
- address[3:2] used to select one of four 32-bit words within a set
- address[1:0] used to select the byte within the 32-bit word

30.3.4 Cache Control

The Code Cache is disabled at reset. Cache tag and data arrays are not cleared at reset. Therefore, to enable the cache, cache commands must be done to clear and initialize the required tag array bits and to configure and enable the caches.

30.3.4.1 Cache set commands

The cache set commands may operate on:

- all of way 0,
- all of way 1, or
- all of both ways (complete cache).

Cache set commands are initiated using the upper bits in the CCR register. Cache set commands perform their operation on the cache independent of the cache enable bit, CCR[ENCACHE].

A cache set command is initiated by setting the CCR[GO] bit. This bit also acts as a busy bit for set commands. It stays set while the command is active and is cleared by the hardware when the set command completes.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Supported cache set commands are given in Table 30-1. Set commands work as follows:

- Invalidate – Unconditionally clear valid and modify bits of a cache entry.
- Push – Push a cache entry if it is valid and modified, then clear the modify bit. If entry not valid or not modified, leave as is.
- Clear – Push a cache entry if it is valid and modified, then clear the valid and modify bits. If entry not valid or not modified, clear the valid bit.

Table 30-1. Cache Set Commands

CCR[27:24]				Command
PUSHW1	INVW1	PUSHW0	INVW0	
0	0	0	0	NOP
0	0	0	1	Invalidate all way 0
0	0	1	0	Push all way 0
0	0	1	1	Clear all way 0
0	1	0	0	Invalidate all way 1
0	1	0	1	Invalidate all way 1; invalidate all way 0 (invalidate cache)
0	1	1	0	Invalidate all way 1; push all way 0
0	1	1	1	Invalidate all way 1; clear all way 0
1	0	0	0	Push all way 1
1	0	0	1	Push all way 1; invalidate all way 0
1	0	1	0	Push all way 1; push all way 0 (push cache)
1	0	1	1	Push all way 1; clear all way 0
1	1	0	0	Clear all way 1
1	1	0	1	Clear all way 1; invalidate all way 0
1	1	1	0	Clear all way 1; push all way 0
1	1	1	1	Clear all way 1; clear all way 0 (clear cache)

After a reset, complete an invalidate cache command before using the cache. It is possible to combine the cache invalidate command with the cache enable. That is, setting CCR to 0x8500_0003 will invalidate the cache and enable the cache and write buffer.

30.3.4.2 Cache line commands

Cache line commands operate on a single line in the cache at a time. Cache line commands can be performed using a physical or cache address.

For Assess

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- A cache address consists of a set address and a way select. The line command acts on the specified cache line.
- Cache line commands with physical addresses first search both ways of the cache set specified by bits [10:4] of the physical address. If they hit, the commands perform their action on the hit way.

Cache line commands are specified using the upper bits in the CLCR register. Cache line commands perform their operation on the cache independent of the cache enable bit (CCR[ENCACHE]). Using a cache address, the command can be completely specified using the CLCR register. Using a physical address, the command must also use the CSAR register to specify the physical address.

A line cache command is initiated by setting the line command go bit (CLCR[LGO] or CSAR[LGO]). This bit also acts a a busy bit for line commands. It stays set while the command is active and is cleared by the hardware when the command completes.

The CLCR[27:24] bits select the line command as follows:

Table 30-2. Cache Line Commands

CLCR[27:24]			Command
LACC	LADSEL	LCMD	
0	0	00	Search by cache address and way
0	0	01	Invalidate by cache address and way
0	0	10	Push by cache address and way
0	0	11	Clear by cache address and way
0	1	00	Search by physical address
0	1	01	Invalidate by physical address
0	1	10	Push by physical address
0	1	11	Clear by physical address
1	0	00	Write by cache address and way
1	0	01	Reserved, NOP
1	0	10	Reserved, NOP
1	0	11	Reserved, NOP
1	1	*xx*	Reserved, NOP

For Internal Purposes Only

30.3.4.2.1 Executing a series of line commands using cache addresses

A series of line commands with incremental cache addresses can be performed by just writing to the CLCR.

- Place the command in CLCR[27:24],
- Set the way (CLCR[WSEL]) and tag/data (CLCR[TDSEL]) controls as needed,

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- Place the cache address in CLCR[CACHEADDR], and
- Set the line command go bit (CLCR[LGO]).

When one line command completes, initiate the next command by following these steps:

- Increment the cache address (at bit 2 to step through data or at bit 4 to step through lines), and
- Set the line command go bit (CLCR[LGO]).

30.3.4.2.2 Executing a series of line commands using physical addresses

Perform a series of line commands with incremental physical addresses using the following steps:

- Write to the CLCR.
  - Place the command in CLCR[27:24]
  - Set the tag/data (CLCR[TDSEL]) control
- Place the physical address in CSAR[PHYADDR] and set the line command go bit (CSAR[LGO]).

When one line command completes, initiate the next command by following these steps:

- Increment the physical address (at bit 2 to step through data or at bit 4 to step through lines), and
- Set the line command go bit (CSAR[LGO]).

The line command go bit is shared between the CLCR and CSAR registers, so that the above steps can be completed in a single write to the CSAR register.

30.3.4.2.3 Line command results

At completion of a line command, the CLCR register contains information on the initial state of the line targeted by the command. For line commands with cache addresses, this information is read before the line command action is performed from the targeted cache line. For line commands with physical addresses, this information is read on a hit before the line command action is performed from the hit cache line or has initial valid bit cleared if the command misses. In general, if the valid indicator (CLCR[LCIVB] is cleared, the targeted line was invalid at the start of the line command and no line operation was performed.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 30-3. Line command results

CLCR[22:20]			For cache address commands	For physical address commands
LCWAY	LCIMB	LCIVB		
0	0	0	Way 0 line was invalid	No hit
0	0	1	Way 0 valid, not modified	Way 0 valid, not modified
0	1	0	Way 0 line was invalid	No hit
0	1	1	Way 0 valid and modified	Way 0 valid and modified
1	0	0	Way 1 line was invalid	No hit
1	0	1	Way 1 valid, not modified	Way 1 valid, not modified
1	1	0	Way 1 line was invalid	No hit
1	1	1	Way 1 valid and modified	Way 1 valid and modified

At completion of a line command other than a write, the CCVR (Cache R/W Value Register) contains information on the initial state of the line tag or data targeted by the command. For line commands, CLCR[TDSEL] selects between tag and data. If the line command used a physical address and missed, the data is don't care. For write commands, the CCVR holds the write data.

For Assessment Purposes

Chapter 31
Miscellaneous System Control Module (MSCM)

31.1 Chip-specific Miscellaneous System Control Module (MSCM)

31.1.1 Chip-specific MSCM information

- TCMU RAM size is 28k.
- TCMU and TCML terms are interchangeable with SRAM_U and SRAM_L

31.2 Overview

The Miscellaneous System Control Module (MSCM) contains CPU configuration registers and on-chip memory controller registers.

31.3 Chip Configuration and Boot

The device configuration is defined by flash test bits, supported memory sizes and packing options. Collectively, these configuration bits define an RCON (reset configuration) value.

Once the core has fetched the needed reset vector(s), it is expected they read core and system configuration information from a globally-accessible slave peripheral that properly converts the information into more appropriate values. More specifically, the core accesses configuration information from a common set of peripheral addresses and the chip configuration logic properly evaluates based on the requesting processor and returns the appropriate value for the given processor, including core identification.

For Assessment Purposes Only

As an example, there is a single 32-bit read-only location for the core identification. A 32-bit read from this location returns a four character ASCII string: 0x43_4D_34 ("Cortex-M4").

The programming model associated with the core configuration information is included as part of the Miscellaneous System Control Module (MSCM). It specifically includes multiple views of the processor configuration; one that is available generically to the core and others that are available to any bus masters in the system.

31.4 MSCM Memory Map/Register Definition

31.4.1 CPU Configuration Memory Map and Registers

The CPU configuration portion of the MSCM module provides a set of memory-mapped read-only addresses defining the processor setup. This portion of the MSCM programming model can only be accessed with privileged mode 32-bit read references; any other access type or size are terminated with an error. If the processor is logically not included in the chip configuration, reads of its configuration registers return zeroes.

The CPU Configuration registers are organized based on the logical processor number (not any type of physical port number) and partitioned into several equal sections:

- Offset addresses 0x000 - 0x01F define the generic processor "x" configuration. This region is only accessible to the processor core(s); reads by non-core bus masters are treated as RAZ (read as zero) accesses.
- Offset addresses 0x020 - 0x03F define the configuration information for processor 0 (CP0). This region is accessible to any bus master.

Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

NOTE

MSCM_OCMDR3 is not available on this device.

MSCM memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Processor X Type Register (MSCM_CPxTYPE)	32	R	Undefined	31.4.2/587
4	Processor X Number Register (MSCM_CPxNUM)	32	R	Undefined	31.4.3/588
8	Processor X Master Register (MSCM_CPxMASTER)	32	R	Undefined	31.4.4/589
C	Processor X Count Register (MSCM_CPxCOUNT)	32	R	Undefined	31.4.5/590
10	Processor 0 Configuration 0 Register (MSCM_CP0CFG0)	32	R	0400_0000h	31.4.6/590

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Assessment Purposes Only

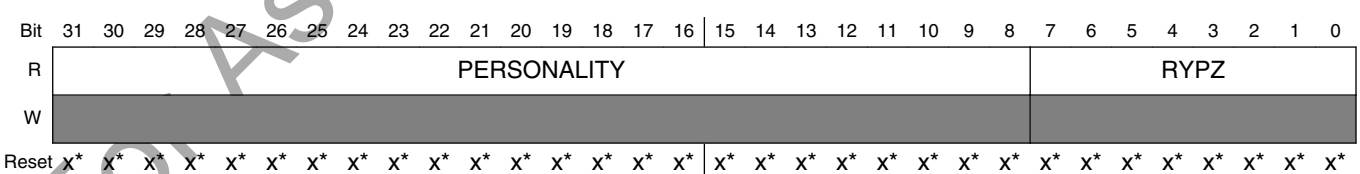
MSCM memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
14	Processor 0 Configuration 1 Register (MSCM_CP0CFG1)	32	R	0000_0000h	31.4.7/592
18	Processor 0 Configuration 2 Register (MSCM_CP0CFG2)	32	R	0701_0701h	31.4.8/593
1C	Processor 0 Configuration 3 Register (MSCM_CP0CFG3)	32	R	0000_0101h	31.4.9/594
20	Processor 0 Type Register (MSCM_CP0TYPE)	32	R	434D_3401h	31.4.10/ 595
24	Processor 0 Number Register (MSCM_CP0NUM)	32	R	0000_0000h	31.4.11/ 596
28	Processor 0 Master Register (MSCM_CP0MASTER)	32	R	0000_0000h	31.4.12/ 597
2C	Processor 0 Count Register (MSCM_CP0COUNT)	32	R	0000_0000h	31.4.13/ 597
400	On-Chip Memory Descriptor Register (MSCM_OCMDR0)	32	R/W	See section	31.4.14/ 598
404	On-Chip Memory Descriptor Register (MSCM_OCMDR1)	32	R/W	See section	31.4.14/ 598
408	On-Chip Memory Descriptor Register (MSCM_OCMDR2)	32	R/W	See section	31.4.14/ 598
40C	On-Chip Memory Descriptor Register (MSCM_OCMDR3)	32	R/W	See section	31.4.14/ 598

31.4.2 Processor X Type Register (MSCM_CPxTYPE)

The register provides a CPU-specific response indicating the personality of the core making the access. The 32 bit response includes 3 ASCII characters defining the CPU type along with a byte defining the logical revision number. The logical revision number follows ARM's rYpZ nomenclature.

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



* Notes:
- x = Undefined at reset.

MSCM_CPxTYPE field descriptions

Field	Description
31–8 PERSONALITY	Processor x Personality

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment F

MSCM_CPxTYPE field descriptions (continued)

Field	Description
	This read-only field defines the processor personality for CPx if CPx = Cortex-M4, then PERSONALITY = 0x43_4D_34 (“CM4”).
RYPZ	Processor x Revision This read-only field defines the processor revision for CPx: 0x01 corresponds to the r0p1 core release. ...

31.4.3 Processor X Number Register (MSCM_CPxNUM)

The register provides a CPU-specific response indicating the logical processor number of the core making the access. In single processor configurations, the logical processor number is always zero.

Address: 0h base + 4h offset = 4h

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

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

* Notes:
- x = Undefined at reset.

MSCM_CPxNUM field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 CPN	Processor x Number This zero-filled word defines the logical processor number for CPx **NOTE:** If single core configuration, then CPN = 0.

Assessment Purposes Only

31.4.4 Processor X Master Register (MSCM_CPxMASTER)

The register provides a CPU-specific response indicating the physical bus master number of the core making the access. The 32 bit response defines the physical master number for processor x.

A privileged read from the CM4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Address: 0h base + 8h offset = 8h

Bit	31	30	29	28	27	26	25	24	23	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																										PPN					
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.

MSCM_CPxMASTER field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PPN	Processor x Physical Port Number This read-only field defines the physical port number for CPUx. **NOTE:** For single core (CPU0), PPN = 0x00.

For Assessment Purposes Only

31.4.5 Processor X Count Register (MSCM_CPxCOUNT)

The register provides a CPU-specific response indicating the total number of processor cores in the chip configuration.

Address: 0h base + Ch offset = Ch

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

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

* Notes:
- x = Undefined at reset.

MSCM_CPxCOUNT field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PCNT	Processor Count This read-only field defines the processor count for the chip configuration: **NOTE:** If single core configuration, then PCNT = 00.

31.4.6 Processor 0 Configuration 0 Register (MSCM_CP0CFG0)

The MSCM_CP0CFG0 register provides a CPU-specific response detailing configuration information, in this case, information on the Level 1 caches (if present).

Access: Privileged read-only

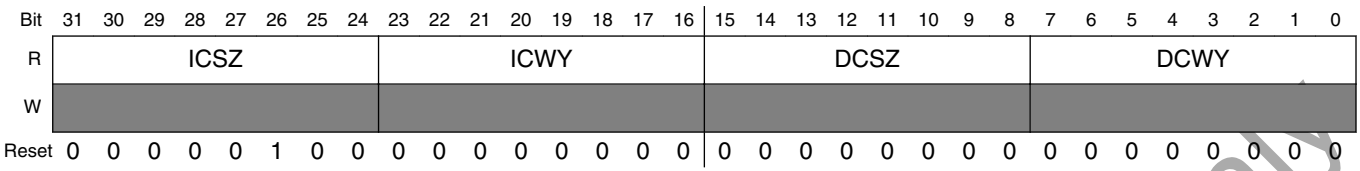
NOTE

The instruction cache is set up to cover accesses from the CM4 code port. This is a multiplex of DCODE and ICODE. This includes all accesses below 0x1FFF_FFFF. It is still

For Assessment Purposes Only

recommended that the software should partition the Data to non-cached address

Address: 0h base + 10h offset = 10h



MSCM_CP0CFG0 field descriptions

Field	Description
31–24 ICSZ	Level 1 Instruction Cache Size. This field provides an encoded value of the Instruction Cache size. The capacity of the memory is expressed as Size [bytes] = 2(8+SZ) where SZ is non-zero; a SZ = 0 indicates the memory is not present. if no Instruction Cache, then ICSZ = 0x00 if a 4 Kbyte Instruction Cache, then ICSZ = 0x04 if an 8 Kbyte Instruction Cache, then ICSZ = 0x05 if a 16 Kbyte Instruction Cache, then ICSZ = 0x06 if a 32 Kbyte Instruction Cache, then ICSZ = 0x07 if a 64 Kbyte Instruction Cache, then ICSZ = 0x08 if a 128 Kbyte Instruction Cache, then ICSZ = 0x09 if a 256 Kbyte Instruction Cache, then ICSZ = 0x0A if a 512 Kbyte Instruction Cache, then ICSZ = 0x0B
23–16 ICWY	Level 1 Instruction Cache Ways. This field provides the number of cache ways for the Instruction Cache. if L1 is present, then ICWY = 0x08 (8-way set-associative) else ICWY = 0x00 (not present) For the Cortex-M4 core in this device, ICWY = 0x00 (not present)
15–8 DCSZ	Level 1 Data Cache Size. This field provides an encoded value of the Data Cache size. The capacity of the memory is expressed as Size [bytes] = 2(8+SZ) where SZ is non-zero; a SZ = 0 indicates the memory is not present. if noData Cache, then DCSZ = 0x00 if a 4 Kbyte Data Cache, then DCSZ = 0x04 if an 8 Kbyte Data Cache, then DCSZ = 0x05 if a 16 Kbyte Data Cache, then DCSZ = 0x06 if a 32 Kbyte Data Cache, then DCSZ = 0x07 if a 64 Kbyte Data Cache, then DCSZ = 0x08 if a 128 Kbyte Data Cache, then DCSZ = 0x09 if a 256 Kbyte Data Cache, then DCSZ = 0x0A if a 512 Kbyte Data Cache, then DCSZ = 0x0B
DCWY	Level 1 Data Cache Ways. This field provides the number of cache ways for the Data Cache. if L1 is present, then DCWY = 0x08 (8-way set-associative)

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MSCM_CP0CFG0 field descriptions (continued)

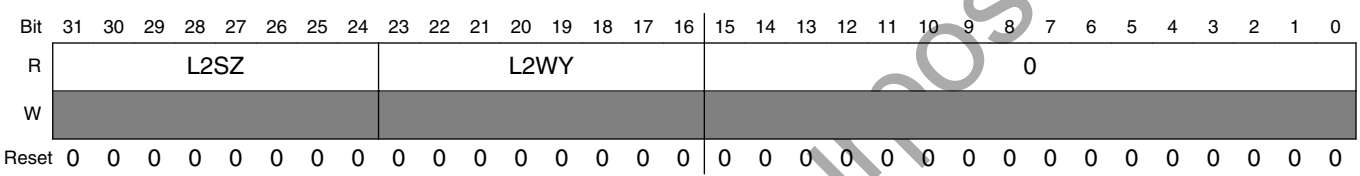
Field	Description
	else DCWY = 0x00 (not present) For the Cortex-M4 core in this device, DCWY = 0x00 (not present)

31.4.7 Processor 0 Configuration 1 Register (MSCM_CP0CFG1)

The MSCM_CP0CFG1 register provides a CPU-specific response detailing configuration information, in this case, information on a Level 2 cache (if present).

Access: Privileged read-only

Address: 0h base + 14h offset = 14h



MSCM_CP0CFG1 field descriptions

Field	Description
31–24 L2SZ	Level 2 Cache Size. This field provides an encoded value of the Level 2 Cache size. The capacity of the memory is expressed as Size [bytes] = 2(8+SZ) where SZ is non-zero; a SZ = 0 indicates the memory is not present. if no Level 2 Cache, then L2SZ = 0x00 if a 4 Kbyte Level 2 Cache, then L2SZ = 0x04 if an 8 Kbyte Level 2 Cache, then L2SZ = 0x05 if a 16 Kbyte Level 2 Cache, then L2SZ = 0x06 if a 32 Kbyte Level 2 Cache, then L2SZ = 0x07 if a 64 Kbyte Level 2 Cache, then L2SZ = 0x08 if a 128 Kbyte Level 2 Cache, then L2SZ = 0x09 if a 256 Kbyte Level 2 Cache, then L2SZ = 0x0A if a 512 Kbyte Level 2 Cache, then L2SZ = 0x0B
23–16 L2WY	Level 2 Cache Ways. This field provides the number of cache ways for the Level 2 Cache. if L2 is present, then L2WY = 0x08 (8-way set-associative) else L2WY = 0x00 (not present) For the Cortex-M4 core in this device, L2WY = 0x00 (not present)
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Freescale Confidential Proprietary Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

31.4.8 Processor 0 Configuration 2 Register (MSCM_CP0CFG2)

The MSCM_CP0CFG2 register provides a CPU-specific response detailing configuration information, in this case, information on tightly-coupled local memories (if present).

Address: 0h base + 18h offset = 18h

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

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

MSCM_CP0CFG2 field descriptions

Field	Description
31–24 TMLSZ	Tightly-coupled Memory Lower Size. This field provides an encoded value of the tightly-coupled local memory lower size. The capacity of the memory is expressed as Size [bytes] = $2^{(8+TMLSZ)}$ where TMLSZ is non-zero; a TMLSZ = 0 indicates the memory is not present. if no TCML, then TMLSZ = 0x00 if a 4 Kbyte TCML, then TMLSZ = 0x04 if an 8 Kbyte TCML, then TMLSZ = 0x05 if a 16 Kbyte TCML, then TMLSZ = 0x06 if a 32 Kbyte TCML, then TMLSZ = 0x07 if a 64 Kbyte TCML, then TMLSZ = 0x08 if a 128 Kbyte TCML, then TMLSZ = 0x09 if a 256 Kbyte TCML, then TMLSZ = 0x0A
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
15–8 TMUSZ	Tightly-coupled Memory Upper Size. This field provides an encoded value of the tightly-coupled local memory upper size. The capacity of the memory is expressed as Size [bytes] = $2^{(8+TMUSZ)}$ where TMUSZ is non-zero; a TMUSZ = 0 indicates the memory is not present. if no TCMU, then TMUSZ = 0x00 if a 4 Kbyte TCMU, then TMUSZ = 0x04 if an 8 Kbyte TCMU, then TMUSZ = 0x05 if a 16 Kbyte TCMU, then TMUSZ = 0x06

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MSCM_CP0CFG2 field descriptions (continued)

Field	Description
	if a 32 Kbyte TCMU, then TMUSZ = 0x07 if a 64 Kbyte TCMU, then TMUSZ = 0x08 if a 128 Kbyte TCMU, then TMUSZ = 0x09 if a 256 Kbyte TCMU, then TMUSZ = 0x0A if a 512 Kbyte TCMU, then TMUSZ = 0x0B
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

31.4.9 Processor 0 Configuration 3 Register (MSCM_CP0CFG3)

The MSCM_CP0CFG3 register provides a CPU-specific response detailing configuration information, in this case, information on processor options.

A privileged read from the Cortex-M4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Access: Privileged read-only

Address: 0h base + 1Ch offset = 1Ch

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						SBP		0	BB	CMP	TZ	MMU	JAZ	SIMD	FPU
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

MSCM_CP0CFG3 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 SBP	System Bus Ports. This field defines the number of physical connections to the system bus fabric for this processor.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MSCM_CP0CFG3 field descriptions (continued)

Field	Description
6 BB	Bit Banding. This field defines if the processor supports "bit banding": if bit banding is not supported, then BB = 0x0 if bit banding is supported, then BB = 0x1
5 CMP	Core Memory Protection unit. This field indicates if the core memory protection hardware is included in the processor. if core memory protection is not included, then CMP = 0x0 if core memory protection is included, then CMP = 0x1
4 TZ	Trust Zone. This field indicates if the Trust Zone capabilities are supported in the processor. if Trust Zone support is not included, then TZ = 0x0 if Trust Zone support is included, then TZ = 0x1
3 MMU	Memory Management Unit. This field indicates if the virtual memory management capabilities are supported in the processor. if MMU support is not included, then MMU = 0x0 if MMU support is included, then MMU = 0x1
2 JAZ	Jazelle. This field indicates if Jazelle hardware is supported in the processor. if Jazelle support is not included, then JAZ = 0x0 if Jazelle support is included, then JAZ = 0x1
1 SIMD	SIMD/NEON instruction support. This field indicates if the instruction set extensions supporting SIMD and/or NEON capabilities are supported in the processor. if SIMD/NEON support is not included, then SIMD = 0x0 if SIMD/NEON support is included, then SIMD = 0x1
0 FPU	Floating Point Unit. This field indicates if hardware support for floating point capabilities are supported in the processor. if FPU support is not included, then FPU = 0x0 if FPU support is included, then FPU = 0x1

For Assess

31.4.10 Processor 0 Type Register (MSCM_CP0TYPE)

The register provides a CPU-specific response indicating the personality of the core making the access. The 32 bit response includes 3 ASCII characters defining the CPU type along with a byte defining the logical revision number. The logical revision number follows ARM's rYpZ nomenclature.

Address: 0h base + 20h offset = 20h

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MSCM_CP0TYPE field descriptions

Field	Description
31–8 PERSONALITY	Processor x Personality This read-only field defines the processor personality for CPx if CPx = Cortex-M4, then PERSONALITY = 0x43_4D_34 (“CM4”).
RYPZ	Processor x Revision This read-only field defines the processor revision for CPx: 0x01 corresponds to the r0p1 core release. ...

31.4.11 Processor 0 Number Register (MSCM_CP0NUM)

The register provides a CPU-specific response indicating the logical processor number of the core making the access. In single processor configurations, the logical processor number is always zero.

Address: 0h base + 24h offset = 24h

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															CPN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MSCM_CP0NUM field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 CPN	Processor x Number This zero-filled word defines the logical processor number for CPx If single core configuration, then CPN = 0

Assessment Purposes Only

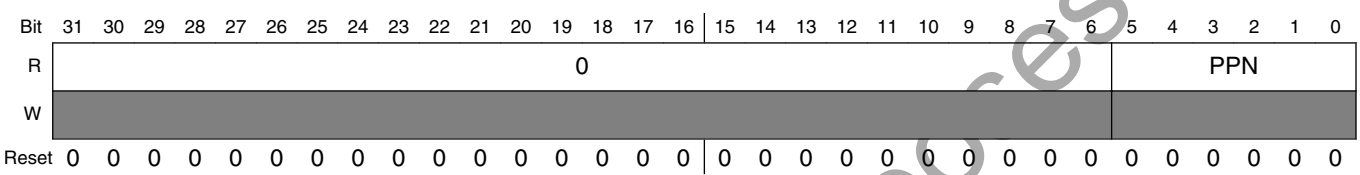
Freescale Confidential Proprietary - Non-Disclosure Agreement required

31.4.12 Processor 0 Master Register (MSCM_CP0MASTER)

The register provides a CPU-specific response indicating the physical bus master number the core making the access. The 32 bit response defines the physical master number for processor x.

A privileged read from the CA5 or the CM4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Address: 0h base + 28h offset = 28h



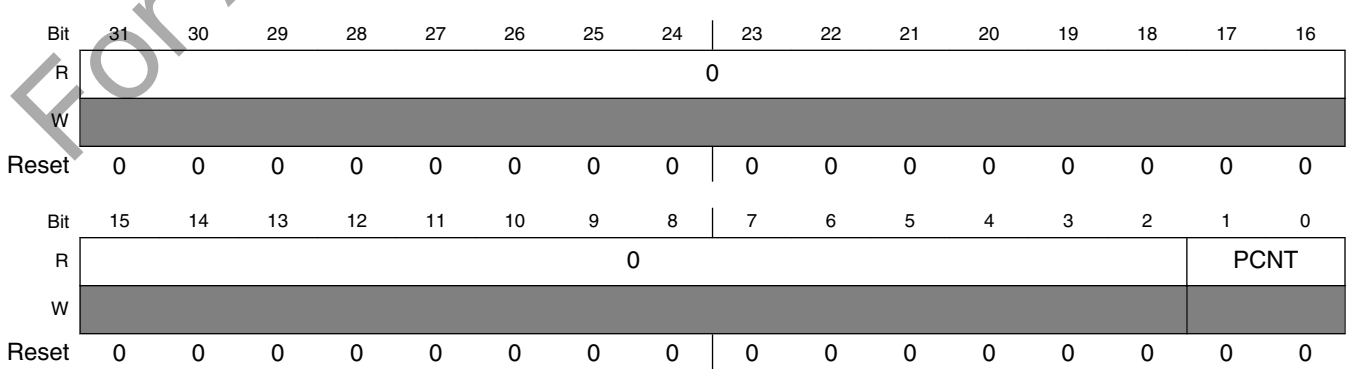
MSCM_CP0MASTER field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PPN	Processor x Physical Port Number This read-only field defines the physical port number for CPUx. For CPU0, PPN = 0x00

31.4.13 Processor 0 Count Register (MSCM_CP0COUNT)

The register provides a CPU-specific response indicating the total number of processor cores in the chip configuration.

Address: 0h base + 2Ch offset = 2Ch



For Assessment Purposes Only

MSCM_CP0COUNT field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PCNT	Processor Count This read-only field defines the processor count for the chip configuration: If single core configuration, then PCNT = 00

31.4.14 On-Chip Memory Descriptor Register (MSCM_OCMDR*n*)

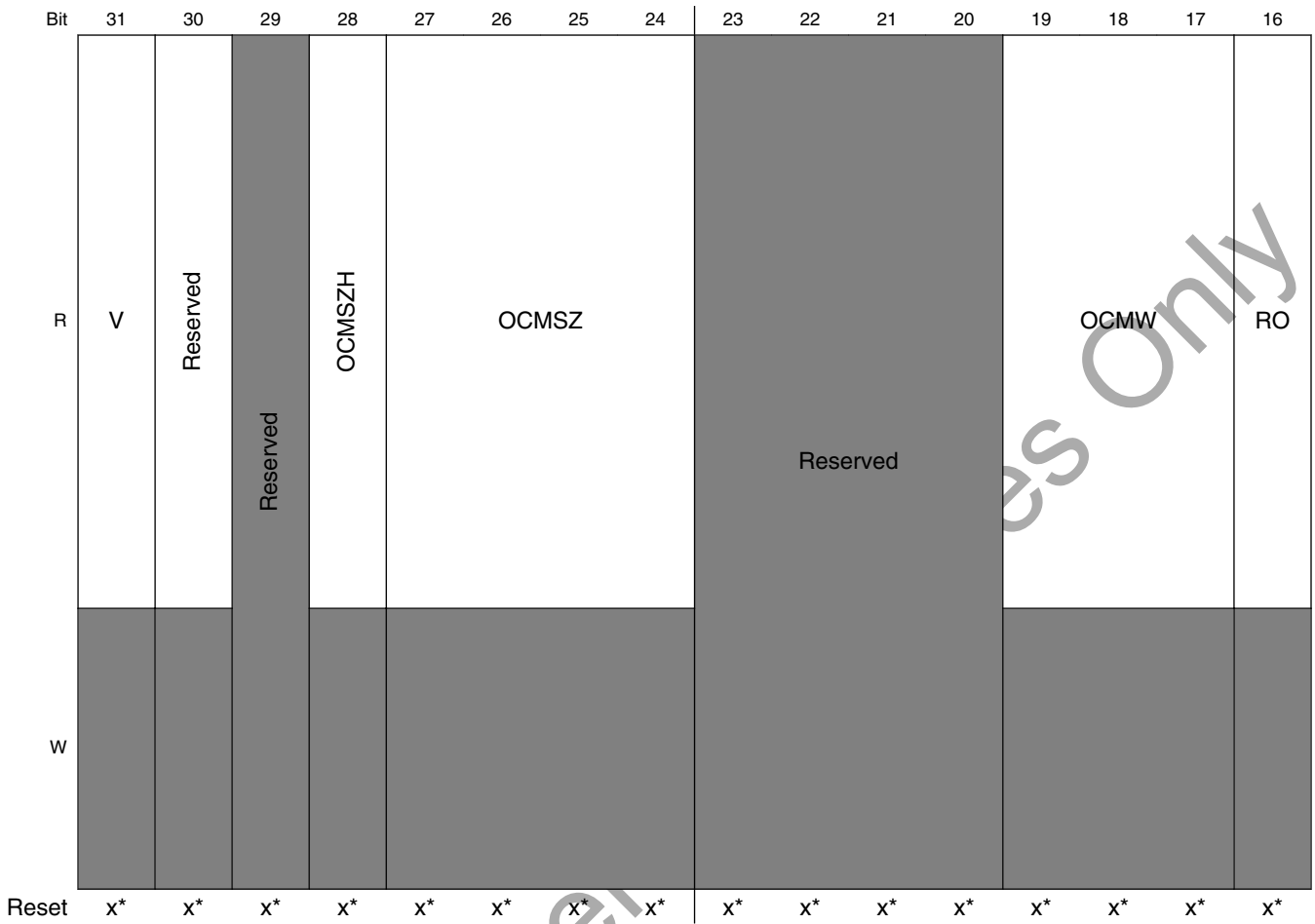
This section of the programming model is an array of 32-bit generic on-chip memory descriptor registers that provide static information on the attached memories as well as configurable controls (where appropriate).

Privileged 32-bit reads from a processor core or the debugger return the appropriate processor information. Reads from any other bus master return all zeroes. Privileged writes from a processor core or the debugger to writeable registers update the appropriate fields. Privileged writes from other bus masters are ignored. Attempted user mode accesses or any access with a size other than 32 bits are terminated with an error.

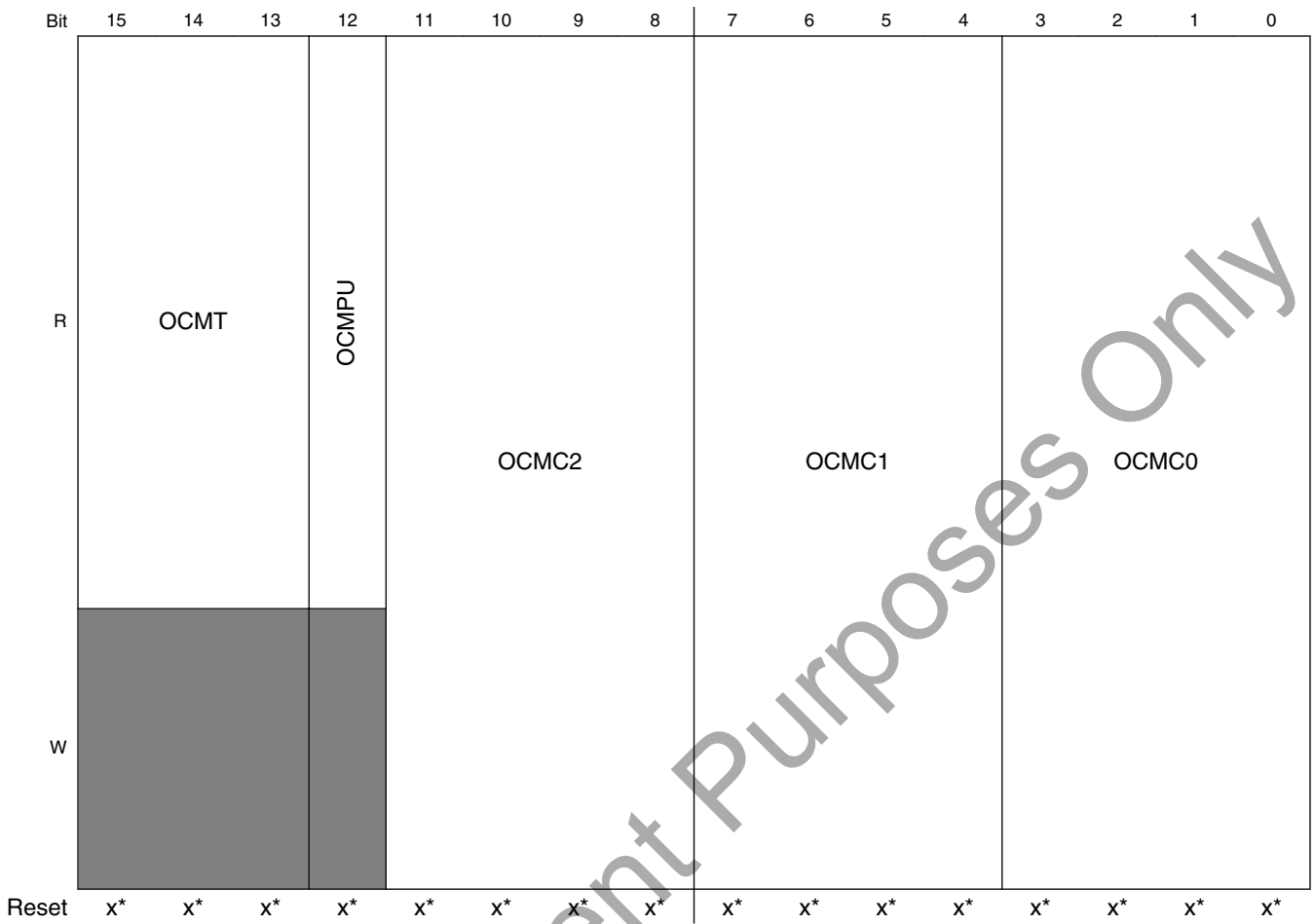
For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 400h offset + (4d × i), where i=0d to 3d



For Assessment Purposes Only



Freescale Confidential Proprietary - Non-Disclosure Agreement required

* Notes:
- OCMDR3 reset value: 0x4000_0000.x = Undefined at reset.

MSCM_OCMDR*n* field descriptions

Field	Description
31 V	OCMEM Valid bit. This read-only field defines the validity (presence) of the on-chip memory 0 OCMEMn is not present. 1 OCMEMn is present.
30 Reserved	This field is reserved.
29 Reserved	This field is reserved.
28 OCMSZH	OCMEM Size “Hole”. For on-chip memories that are not fully populated, that is, include a memory “hole” in the upper 25% of the address range, this bit is used. 0 OCMEMn is a power-of-2 capacity. 1 OCMEMn is not a power-of-2, with a capacity is 0.75 * OCMSZ.
27–24 OCMSZ	OCMEM Size. This read-only field provides an encoded value of the on-chip memory size. The capacity of the memory is expressed as Size [bytes] = $2^{(9+OCMSZ)}$ where OCMSZ is non-zero; a OCMSZ = 0 indicates the memory is not present.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MSCM_OCMDR*n* field descriptions (continued)

Field	Description
	0000 no OCMEMn 0011 4KB OCMEMn 0100 8KB OCMEMn 0101 16KB OCMEMn 0110 32KB OCMEMn 0111 64KB OCMEMn 1000 128KB OCMEMn 1001 256KB OCMEMn 1010 512KB OCMEMn 1011 1024KB OCMEMn 1100 2048KB OCMEMn 1101 4096KB OCMEMn 1110 8192KB OCMEMn 1111 16384KB OCMEMn
23–20 Reserved	This field is reserved.
19–17 OCMW	OCMEM datapath Width. This read-only field defines the width of the on-chip memory: 000-001 Reserved 010 OCMEMn 32-bits wide 011 OCMEMn 64-bits wide 100 OCMEMn 128-bits wide 101 OCMEMn 256-bits wide 110-111 Reserved
16 RO	Read-Only. This register bit provides a mechanism to “lock” the configuration state defined by OCMDRn[11:0]. Once asserted, attempted writes to the OCMDRn[11:0] register are ignored until the next reset clears the flag. 0 writes to the OCMDRn[11:0] are allowed 1 writes to the OCMDRn[11:0] are ignored
15–13 OCMT	OCMEM Type. This field defines the type of the on-chip memory: 000 Reserved 001 Reserved 010 Reserved 011 Reserved 100 OCMEMn is a program flash. 101 OCMEMn is a data flash. 110 OCMEMn is an EEE. 111 Reserved
12 OCMPU	OCMEM Memory Protection Unit. This read-only field identifies a memory protected by a Memory Protection Unit. 0 OCMEMn is not protected by an MPU. 1 OCMEMn is protected by an MPU.
11–8 OCMC2	OCMEM Control Field 2. This 4-bit field (if used) defines the configuration of the on-chip memory. The field’s functionality is dependent on the OCMT value.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

MSCM_OCMDR*n* field descriptions (continued)

Field	Description
7–4 OCMC1	OCMEM Control Field 1. This 4-bit field (if used) defines the configuration of the on-chip memory. The field's functionality is dependent on the OCMT value. OCMDR0[5] or OCMDR1[5] bit controls whether prefetches (or speculative accesses) are initiated in response to instruction fetches or data references, see Speculative reads. • OCMDR bit 4: data prefetch. Value 0 means enable and value 1 means disable. • OCMDR bit 5: flash speculate. Value 0 means enable and value 1 means disable. **NOTE:** The control bit is only applicable when OCMT = 100 (program flash) or 101 (data flash). In other cases, it is unused.
OCMC0	OCMEM Control Field 0. This 4-bit field (if used) defines the configuration of the on-chip memory. The field's functionality is dependent on the OCMT value.

For Assessment Purposes

Chapter 32
Flash Memory Controller (FMC)

32.1 Introduction

The Flash Memory Controller (FMC) is a memory acceleration unit that 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.

32.1.1 Overview

The Flash Memory Controller manages the interface between the device and the dual-bank flash memory. The Program Flash is referred as Bank 0 and the Data Flash is referred as Bank 1. The FMC receives status information detailing the configuration of the memory and uses this information to ensure a proper interface. The following 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.

In addition, for bank 0 and bank 1, the FMC provides separate mechanisms for accelerating the interface between the device and the flash memory. A 128 (64 for bank1, Data Flash)-bit speculation buffer can prefetch the next 128 (64 for bank1, Data Flash)-bit flash memory location, and a single-entry 128 (64 for bank1, Data Flash)-bit buffer can store previously accessed flash memory or FlexNVM data for quick access times.

For Assessment Purposes Only

32.1.2 Features

The FMC's features include:

- Interface between the device and the dual-bank flash memory and FlexMemory:
  - 8-bit, 16-bit, and 32-bit read operations to program flash memory and FlexNVM used as data flash memory.
  - 8-bit, 16-bit, and 32-bit read and write operations to FlexNVM and FlexRAM used as EEPROM.
  - For bank 0 (Program Flash): The memory returns 128 bits, therefore read accesses to consecutive 32-bit spaces in memory return the second, third, and fourth read data with no wait states. For bank 1 (Data Flash): The memory returns 64 bits, therefore read accesses to consecutive 32-bit spaces in memory return the second read data with no wait states.
- For bank 0 and bank 1: Acceleration of data transfer from program flash memory and FlexMemory to the device:
  - 128 (64 for bank1, Data Flash)-bit prefetch speculation buffer with controls for instruction/data access per master and bank
  - Single-entry buffer per bank
  - Invalidation control for the speculation buffer and the single-entry buffer

32.2 Modes of operation

The FMC only operates when a bus master accesses the flash memory or FlexMemory.

In terms of device power modes, the FMC only operates in run and wait modes, including VLPR and VLPW modes.

For any device power mode where the flash memory or FlexMemory cannot be accessed, the FMC is disabled.

32.3 External signal description

The FMC has no external signals.

32.4 Functional description

The FMC is a flash acceleration unit with flexible buffers for user configuration. Whenever a hit occurs for the prefetch speculation buffer, or the single-entry buffer, the requested data is transferred within a single system clock.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

32.4.1 Default configuration

Upon system reset, the FMC is configured to provide a significant level of buffering for transfers from the flash memory or FlexMemory:

- For bank 0 and bank 1:
  - Prefetch support for data and instructions is enabled.
  - The single-entry buffer is enabled.

32.4.2 Speculative reads

The FMC has a single buffer that reads ahead to the next word in the flash memory if there is an idle cycle. Speculative prefetching is programmable for each bank for instruction and/or data accesses using MSCM_OCMDR0[5] and MSCM_OCMDR1[5] (value 0 means "enables prefetches (or speculative 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 address after a read completes. By requesting the next word immediately, speculative reads can help to reduce or even eliminate wait states when accessing sequential code and/or data.

For example, consider the following scenario:

- Assume a system with a 4:1 core-to-flash clock ratio and with speculative reads enabled.
- The core requests four (for Data Flash, bank 1) or eight (for Program Flash, bank 0) sequential longwords in back-to-back requests, meaning there are no core cycle delays except for stalls waiting for flash memory data to be returned.
- None of the data is already stored in the cache or speculation buffer.

In this scenario, the sequence of events for accessing the four (for Data Flash, bank 1) oreight (for Program Flash, bank 0) longwords is as follows:

1. The first longword read requires 4 to 7 core clocks.
2. Due to the 128-bit data bus of the flash memory, the second longword read takes only 1 core clock because the data is already available inside the FMC. For the same reason, the third and fourth longword reads each take only 1 core clock.
3. Per 64-bit for Data Flash (bank 1), accessing the third longword requires 3 core clock cycles. The flash memory read itself takes 4 clocks, but the first clock overlaps with the second longword read.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

4. Per 128-bit for Program Flash (bank 0), accessing the fifth longword requires 1 core clock cycle. The flash memory read itself takes 4 clocks, but the access starts immediately after the first read. As a result, 3 clocks for this access overlap with the second, third, and fourth longword reads from the core.
5. Per 64-bit for Data Flash (bank 1), reading the fourth longword, like the second longword, takes only 1 clock due to the 64-bit flash memory data bus.
6. Per 128-bit for Program Flash (bank 0), reading the sixth, seventh, and eighth longwords takes only 1 clock each because the data is already available inside the FMC.

32.5 Initialization and application information

The FMC does not require user initialization. Flash acceleration features are enabled by default.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 33
Flash Memory Module (FTFE)

33.1 Introduction

The FTFE module includes the following accessible memory regions:

- Program flash memory for vector space and code store
- FlexNVM for data store and additional code store
- FlexRAM for high-endurance data store or traditional RAM

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.

For Assessment Purposes Only

33.1.1 Features

The FTFE module includes the following features.

NOTE

See the device's Chip Configuration details for the exact amount of flash memory available on your device.

33.1.1.1 Program Flash Memory Features

- Sector size of 4 Kbytes
- Program flash 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 program flash block is possible while programming or erasing data in the data flash block or FlexRAM

33.1.1.2 FlexNVM memory features

When FlexNVM is partitioned for data flash 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

33.1.1.3 FlexRAM features

- Memory that can be used as traditional RAM or as high-endurance EEPROM storage
- Up to 4 Kbytes of FlexRAM configured for EEPROM or traditional RAM operations
- When configured for EEPROM:

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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

33.1.1.4 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

33.1.2 Block diagram

The block diagram of the FTFE module is shown in the following figure.

For Assessment Purposes Only

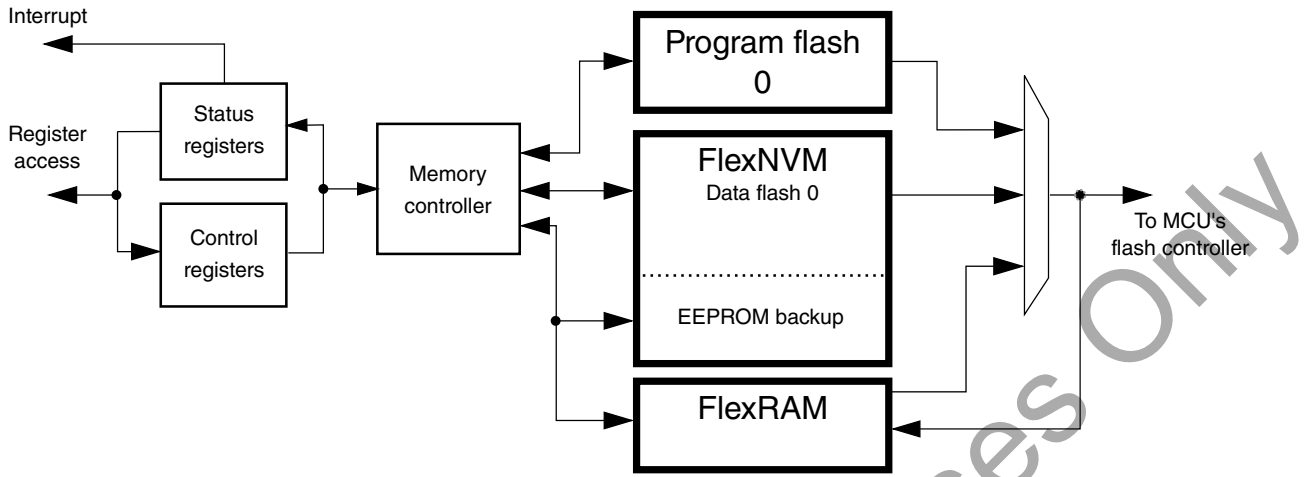


Figure 33-1. FTFE block diagram

33.1.3 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

HSRUN — An MCU power mode enabling high-speed access to the memory resources in the FTFE module. The user has no access to the Flash command set when the MCU is in HSRUN mode.

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.

NVM Special Mode — An NVM mode enabling external, off-chip access to the memory resources in the FTFE module. A reduced flash command set is available when the MCU is secured. See the Chip Configuration details for information on when this mode is used.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Double-Phrase — 128 bits of data with an aligned double-phrase having byte-address[3:0] = 0000.

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 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.

33.2 External signal description

The FTFE module contains no signals that connect off-chip.

33.3 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.3.1 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	Data flash protection byte. Refer to the description of the Data Flash Protection Register (FDPROT).
0x0_040E	1	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).

33.3.2 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 – 0x3BF	960	Reserved
0x3C0 – 0x3FF	64	Program Once Field

For Assessment Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.3.2.1 Program Once field

The Program Once field in the program flash 0 IFR provides 64 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).

33.3.3 Data flash 0 IFR map

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

33.3.3.1 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 and indicates whether the FlexRAM is loaded with valid EEPROM data during the flash reset sequence. To program the EEERST, EEESIZE value, see the Program Partition command described in Program Partition command.

Table 33-1. EEPROM Data Set Size

Data flash IFR: 0x03FD							
7	6	5	4	3	2	1	0
1	EEERST	EEESPLIT		EEESIZE			
	= Unimplemented or Reserved						

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 33-2. 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' = Reserved '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

For A

33.3.3.2 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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 33-3. FlexNVM partition code

Data Flash IFR: 0x03FC							
7	6	5	4	3	2	1	0
1	1	1	1	DEPART			
	= Unimplemented or Reserved						

Table 33-4. 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	64	0
0001	Reserved	Reserved
0010	Reserved	Reserved
0011	32	32
0100	0	64
0101	Reserved	Reserved
0110	Reserved	Reserved
0111	Reserved	Reserved
1000	0	64
1001	Reserved	Reserved
1010	16	48
1011	32	32
1100	64	0
1101	Reserved	Reserved
1110	Reserved	Reserved
1111	64	0

For A

33.3.4 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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Flash Status Register (FTFE_FSTAT)	8	R/W	00h	33.3.4.1/ 618
1	Flash Configuration Register (FTFE_FCNFG)	8	R/W	00h	33.3.4.2/ 619
2	Flash Security Register (FTFE_FSEC)	8	R	Undefined	33.3.4.3/ 621
3	Flash Option Register (FTFE_FOPT)	8	R	Undefined	33.3.4.4/ 622
4	Flash Common Command Object Registers (FTFE_FCCOB3)	8	R/W	00h	33.3.4.5/ 623
5	Flash Common Command Object Registers (FTFE_FCCOB2)	8	R/W	00h	33.3.4.5/ 623
6	Flash Common Command Object Registers (FTFE_FCCOB1)	8	R/W	00h	33.3.4.5/ 623
7	Flash Common Command Object Registers (FTFE_FCCOB0)	8	R/W	00h	33.3.4.5/ 623
8	Flash Common Command Object Registers (FTFE_FCCOB7)	8	R/W	00h	33.3.4.5/ 623
9	Flash Common Command Object Registers (FTFE_FCCOB6)	8	R/W	00h	33.3.4.5/ 623
A	Flash Common Command Object Registers (FTFE_FCCOB5)	8	R/W	00h	33.3.4.5/ 623
B	Flash Common Command Object Registers (FTFE_FCCOB4)	8	R/W	00h	33.3.4.5/ 623
C	Flash Common Command Object Registers (FTFE_FCCOBB)	8	R/W	00h	33.3.4.5/ 623
D	Flash Common Command Object Registers (FTFE_FCCOBA)	8	R/W	00h	33.3.4.5/ 623
E	Flash Common Command Object Registers (FTFE_FCCOB9)	8	R/W	00h	33.3.4.5/ 623
F	Flash Common Command Object Registers (FTFE_FCCOB8)	8	R/W	00h	33.3.4.5/ 623
10	Program Flash Protection Registers (FTFE_FPROT3)	8	R/W	Undefined	33.3.4.6/ 624
11	Program Flash Protection Registers (FTFE_FPROT2)	8	R/W	Undefined	33.3.4.6/ 624
12	Program Flash Protection Registers (FTFE_FPROT1)	8	R/W	Undefined	33.3.4.6/ 624

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTFE memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
13	Program Flash Protection Registers (FTFE_FPROT0)	8	R/W	Undefined	33.3.4.6/ 624
16	EEPROM Protection Register (FTFE_FEPROT)	8	R/W	Undefined	33.3.4.7/ 626
17	Data Flash Protection Register (FTFE_FDPROT)	8	R/W	Undefined	33.3.4.8/ 627
2E	Flash Error Status Register (FTFE_FERSTAT)	8	R/W	00h	33.3.4.9/ 628
2F	Flash Error Configuration Register (FTFE_FERCNFG)	8	R/W	00h	33.3.4.10/ 629

33.3.4.1 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: 0h base + 0h offset = 0h

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 EEE, 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.

Table continues on the next page...

Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTFE_FSTAT field descriptions (continued)

Field	Description
	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. 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.

For r

33.3.4.2 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. PFLSH, RAMRDY, and EEERDY are read-only status bits. The reset values for the PFLSH, RAMRDY, and EEERDY bits are determined during the reset sequence.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 1h offset = 1h

Bit	7	6	5	4	3	2	1	0
Read	CCIE	RDCOLLIE	ERSAREQ	ERSSUSP	0	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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 PFLSH	FTFE configuration 0 FTFE configuration supports one program flash blocks and one FlexNVM block 1 Reserved
1 RAMRDY	RAM Ready This flag indicates the current status of the FlexRAM.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTFE_FCNFG field descriptions (continued)

Field	Description
	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 with the option to load the FlexRAM during the reset sequence and will be set if the FlexNVM block is not partitioned for EEPROM or if the FlexNVM block is partitioned for EEPROM with the option to not load the FlexRAM during the reset sequence. 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. 0 FlexRAM is not available for traditional RAM access 1 FlexRAM is available as traditional RAM only; writes to the FlexRAM do not trigger EEPROM operations
0 EEERDY	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. 0 FlexRAM is not available for EEPROM operation 1 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

33.3.4.3 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: 0h base + 2h offset = 2h

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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Pu

FTFE_FSEC field descriptions (continued)

Field	Description
	00 Backdoor key access disabled 01 Backdoor key access disabled (preferred KEYEN state to disable backdoor key access) 10 Backdoor key access enabled 11 Backdoor key access disabled
5–4 MEEN	Mass Erase Enable Bits Enables and disables mass erase capability of the FTFE module. The state of the MEEN bits is only relevant when the SEC bits are set to secure. 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

For As

33.3.4.4 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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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: 0h base + 3h offset = 3h

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.

33.3.4.5 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: 0h 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

Field	Description
CCOBn	The FCCOB register provides a command code and relevant parameters to the memory controller. The individual registers that compose the FCCOB data set can be written in any order, but you must provide all needed values, which vary from command to command. First, set up all required FCCOB fields and then initiate the command's execution by writing a 1 to the FSTAT[CCIF] bit. This clears the CCIF bit, which locks all FCCOB parameter fields and they cannot be changed by the user until the command completes (CCIF returns to 1). No command buffering or queueing is provided; the next command can be loaded only after the current command completes. Some commands return information to the FCCOB registers. Any values returned to FCCOB are available for reading after the FSTAT[CCIF] flag returns to 1 by the memory controller.

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTFE_FCCOB*n* field descriptions (continued)

Field	Description
	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.

FCCOB Number[1]	Typical Command Parameter Contents [7:0]
0	FCMD (a code that defines the FTFE command)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0]
4	Data Byte 0
5	Data Byte 1
6	Data Byte 2
7	Data Byte 3
8	Data Byte 4
9	Data Byte 5
A	Data Byte 6
B	Data Byte 7

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).

1. Refers to FCCOB register name, not register address

For Asse

33.3.4.6 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]

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Program flash protection register	Program flash protection bits
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: 0h 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 Each program flash region can be protected from program and erase operations by setting the associated PROT bit to the protected state. **In NVM Normal mode:** 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. **In NVM Special mode:** All bits of FPROT are writable without restriction. Unprotected areas can be protected and protected areas can be unprotected. **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.

ment Purpos ly

FTFE_FPROT*n* field descriptions (continued)

Field	Description
	0 Program flash region is protected. 1 Program flash region is not protected

33.3.4.7 EEPROM Protection Register (FTFE_FEPROT)

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.

Address: 0h base + 16h offset = 16h

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 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). **In NVM Normal mode:** 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. **In NVM Special mode:** All bits of the FEPROT register are writable without restriction. Unprotected areas can be protected and protected areas can be unprotected. **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 FlexRAM 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.

Content Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTFE_FEPROT field descriptions (continued)

Field	Description
	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 EEPROM region is protected 1 EEPROM region is not protected

33.3.4.8 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: 0h base + 17h offset = 17h

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 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. **In NVM Normal mode:** 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. **In NVM Special mode:** All bits of the FDPROT register are writable without restriction. Unprotected areas can be protected and protected areas can be unprotected. **Restriction:** The user must never write to the FDPROT register while a command is running (CCIF=0).

nent Purposes Only

FTFE_FDPROT field descriptions (continued)

Field	Description
	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

Content Purposes

33.3.4.9 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: 0h base + 2Eh offset = 2Eh

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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.3.4.10 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: 0h base + 2Fh offset = 2Fh

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.

33.4 Functional Description

The following sections describe functional details of the FTFE module.

Freescale Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.4.1 Flash Protection

Individual regions within the flash memory can be protected from program and erase operations. Protection is controlled by the following registers:

- FPROT*n* — Four registers protect 32 regions of the program flash memory as shown in the following figure

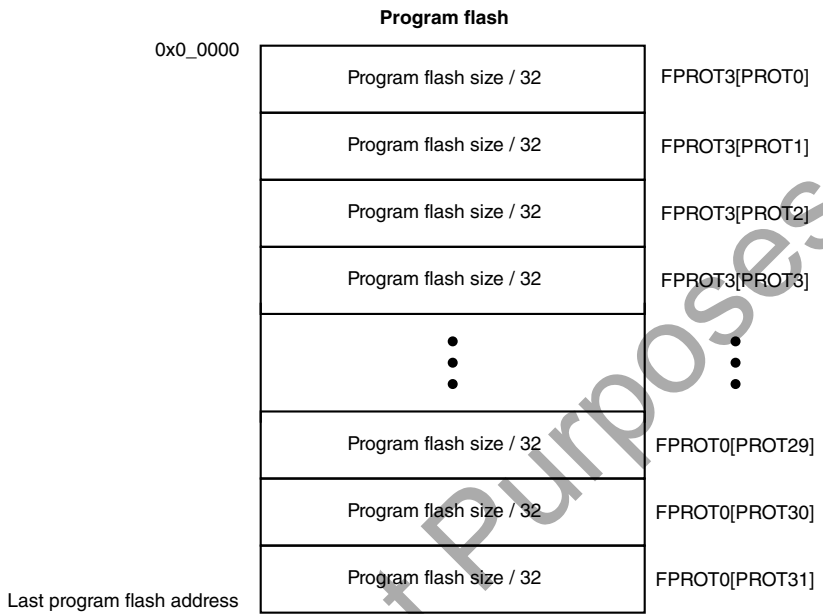


Figure 33-2. Program flash protection

- FDPROT —
  - For 2^n data flash sizes, protects eight regions of the data flash memory as shown in the following figure

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

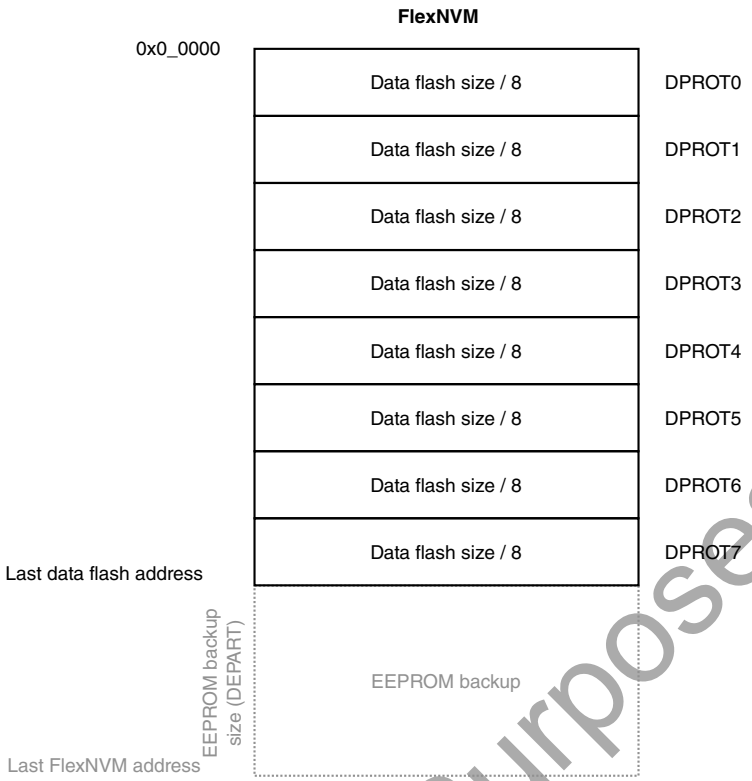


Figure 33-3. Data flash protection (2^n data flash sizes)

- FEPROT — Protects eight regions of the EEPROM memory as shown in the following figure

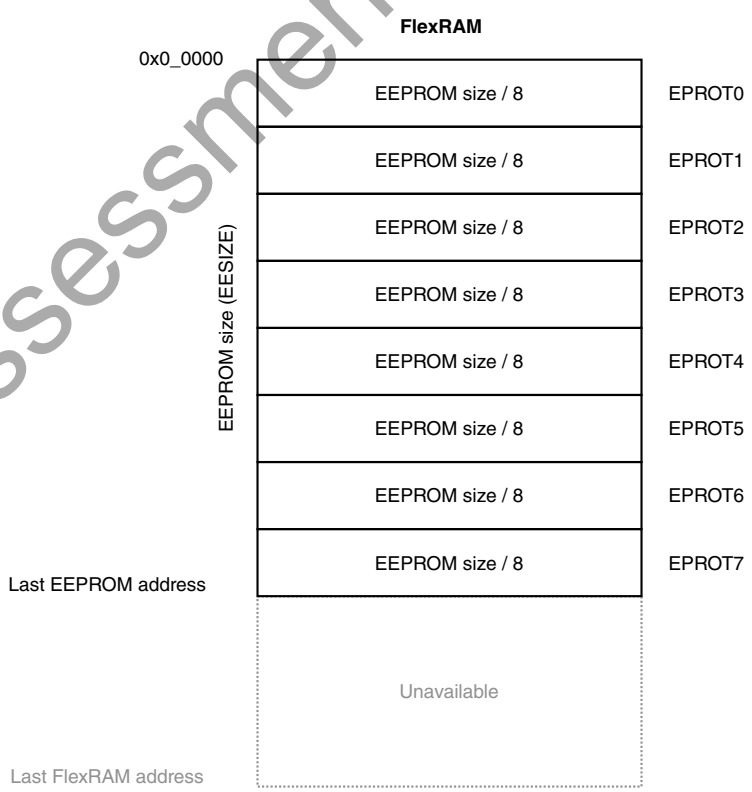


Figure 33-4. EEPROM protection

For Assessment Purposes Only

33.4.2 FlexNVM Description

This section describes the FlexNVM memory.

33.4.2.1 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.

33.4.2.2 EEPROM User Perspective

The EEPROM system is shown in the following figure.

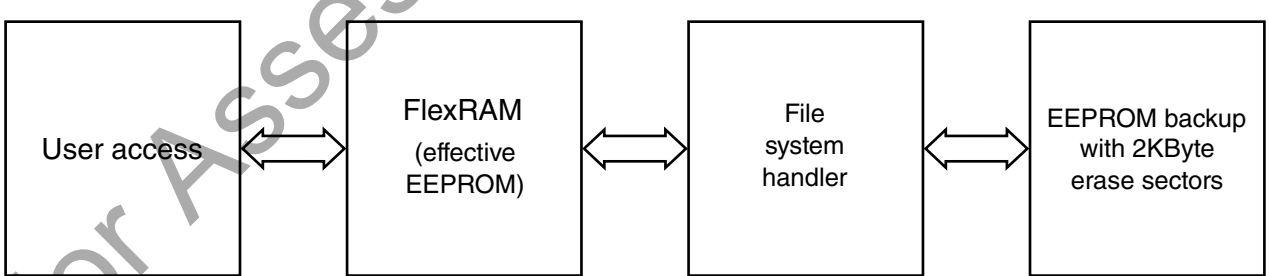


Figure 33-5. Top Level EEPROM Architecture

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 33-2). The remainder of the FlexRAM not used for EEPROM is not accessible while

For Assessment Purposes Only

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 33-4).
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.

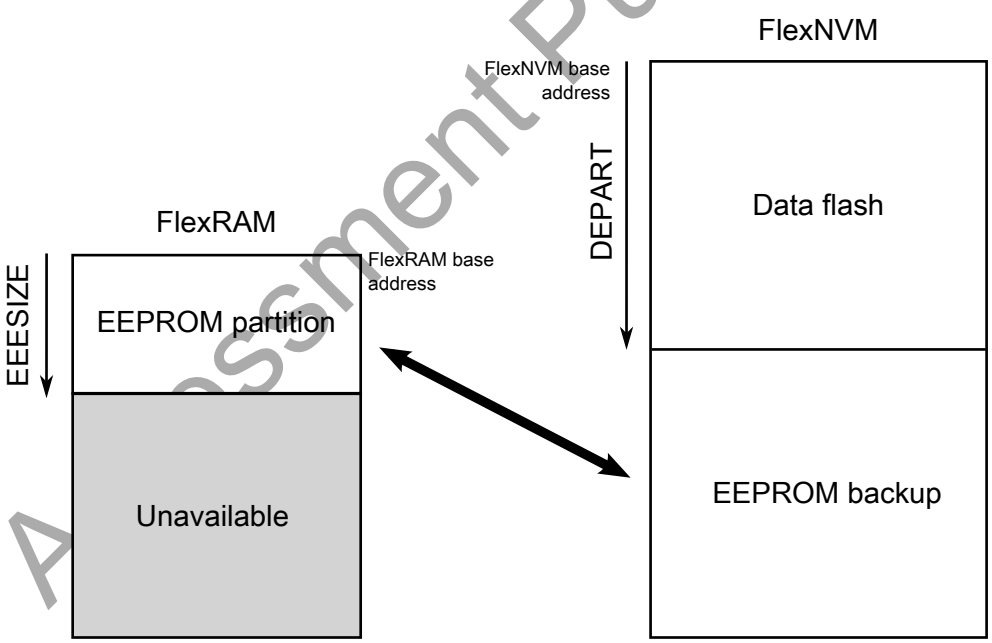


Figure 33-6. FlexRAM to FlexNVM Memory Mapping for EEPROM

For Assessment Purposes Only

33.4.2.3 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.

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.

33.4.2.4 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.

33.4.3 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 33-5. 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.

33.4.4 Flash Operation in Low-Power Modes

33.4.4.1 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).

33.4.4.2 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.

33.4.5 Functional modes of operation

The FTFE module has two operating modes: NVM Normal and NVM Special. The operating mode affects the command set availability (see Table 33-6). Refer to the Chip Configuration details of this device for how to activate each mode.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.4.6 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).

33.4.7 Read while write (RWW)

The following simultaneous accesses are allowed:

- 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 for EEE, are not possible.

Simultaneous operations are further discussed in Allowed simultaneous flash operations.

33.4.8 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.

For Assessment Purposes Only

33.4.9 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

33.4.9.1 Command Write Sequence

FTFE commands are specified using a command write sequence illustrated in Figure 33-7. 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 an FTFE command in HSRUN mode will be trapped with the ACCERR flag being set.

33.4.9.1.1 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.

33.4.9.1.2 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.4.9.1.3 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

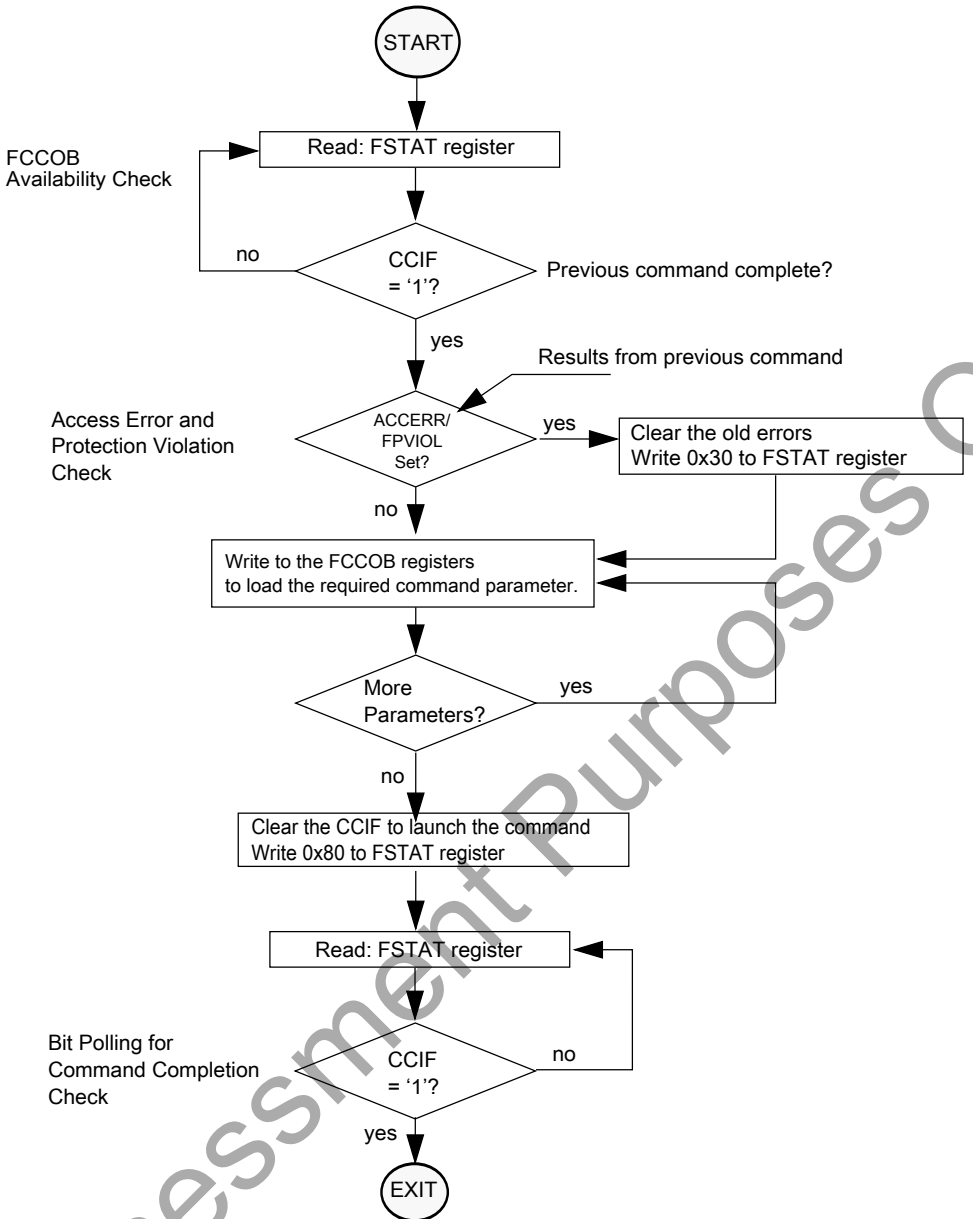


Figure 33-7. Generic Flash Command Write Sequence Flowchart

For Assessment Purposes Only

33.4.9.2 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	Data flash	FlexRAM	Function
0x00	Read 1s Block	×	×		Verify that a program flash or data flash block is erased. FlexNVM

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FCMD	Command	Program flash 0	Data flash	FlexRAM	Function
					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		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.
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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FCMD	Command	Program flash 0	Data flash	FlexRAM	Function
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, 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 all memory locations are unprotected.
0x45	Verify Backdoor Access Key	×			Release MCU security after comparing a set of user-supplied security keys to those stored in the program flash.
0x49	Erase All Blocks Unsecure	×	×	×	Erase all program flash blocks, 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.
0x80	Program Partition		IFR, ×	×	Program the FlexNVM Partition Code and EEPROM 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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FCMD	Command	Program flash 0	Data flash	FlexRAM	Function
					EEPROM. When switching to EEPROM, FlexNVM is not available while valid data records are being copied from EEPROM backup to FlexRAM.

33.4.9.3 Flash commands by mode

The following table shows the flash commands that can be executed in each flash operating mode.

Table 33-6. Flash commands by mode

FCMD	Command	NVM Normal			NVM Special		
		Unsecure	Secure	MEEN=10	Unsecure	Secure	MEEN=10
0x00	Read 1s Block	×	×	×	×	—	—
0x01	Read 1s Section	×	×	×	×	—	—
0x02	Program Check	×	×	×	×	—	—
0x03	Read Resource	×	×	×	×	—	—
0x07	Program Phrase	×	×	×	×	—	—
0x08	Erase Flash Block	×	×	×	×	—	—
0x09	Erase Flash Sector	×	×	×	×	—	—
0x0B	Program Section	×	×	×	×	—	—
0x40	Read 1s All Blocks	×	×	×	×	×	—
0x41	Read Once	×	×	×	×	—	—
0x43	Program Once	×	×	×	×	—	—
0x44	Erase All Blocks	×	×	×	×	×	—
0x45	Verify Backdoor Access Key	×	×	×	×	—	—
0x49	Erase All Blocks Unsecure	×	×	—	×	×	—
0x80	Program Partition	×	×	×	×	—	—
0x81	Set FlexRAM Function	×	×	×	×	—	—

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.4.9.4 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.

Table 33-7. 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.

33.4.10 Margin Read Commands

The Read-1s commands (Read 1s All Blocks, Read 1s Block, Read 1s Section) and the Program Check command have a margin choice parameter that allows the user to apply non-standard read reference levels to the program flash 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.

For Assess Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

33.4.11 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

33.4.11.1 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 33-8. 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 128-bit aligned (Flash address [3:0] = 0000) for program flash, 64-bit aligned (Flash address [2:0] = 000) for data flash.

After clearing CCIF to launch the Read 1s Block command, the FTFE sets the read margin for 1s according to Table 33-9 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.

For Assessment Purposes Only

Table 33-9. 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 33-10. 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 128-bit aligned for program flash, 64-bit aligned for data flash	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

33.4.11.2 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 double-phrases to be verified for program flash, phrases for data flash.

Table 33-11. Read 1s Section Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x01 (RD1SEC)
1	Flash address [23:16] of the first double-phrase to be verified for program flash, phrase for data flash
2	Flash address [15:8] of the first double-phrase to be verified for program flash, phrase for data flash
3	Flash address [7:0][1] of the first double-phrase to be verified for program flash, phrase for data flash
4	Number of doublephrases to be verified for program flash, phrases for data flash [15:8]
5	Number of doublephrases to be verified for program flash, phrases for data flash [7:0]
6	Read-1 Margin Choice

1. Must be 128-bit aligned (Flash address [3:0] = 0000) for program flash, 64-bit aligned (Flash address [2:0] = 000) for data flash.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Upon clearing CCIF to launch the Read 1s Section command, the FTFE sets the read margin for 1s according to Table 33-12 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.

Table 33-12. 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 33-13. 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 128-bit aligned for program flash, 64-bit aligned for data flash	FSTAT[ACCERR]
The requested section crosses a flash block boundary	FSTAT[ACCERR]
The requested number of double-phrases for program flash, phrases for data flash is zero	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

33.4.11.3 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 33-14. 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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 33-15. 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 33-15. 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 33-16. 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]
Either of the margin reads does not match the expected data	FSTAT[MGSTAT0]

Freescale Confidential Proprietary Non-Disclosure Agreement required Purposes Only

33.4.11.4 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 33-17. 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 33-18)
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]
B	Read Data [7:0]

1. Must be 64-bit aligned (Flash address [2:0] = 000).

Table 33-18. 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	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 33-19. Read Resource Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 33-19. Read Resource Command Error Handling (continued)

Error Condition	Error Bit
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]

33.4.11.5 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 33-20. 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. The targeted flash locations must be currently unprotected (see the description of the FPROT registers) to permit execution of the Program Phrase operation.

For Purposes Only

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 33-21. 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]
Any errors have been encountered during the verify operation.	FSTAT[MGSTAT0]

33.4.11.6 Erase Flash Block Command

The Erase Flash Block operation erases all addresses in a single program flash or data flash block.

Table 33-22. 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 128-bit aligned (Flash address [3:0] = 0000) for program flash, 64-bit aligned (Flash address [2:0] = 000) for data flash.

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 33-4) 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

For Development Purpose Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

data flash protection (FDPROT) registers). 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 33-23. 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]
Flash address is not 128-bit aligned for program flash, 64-bit aligned for data flash	FSTAT[ACCERR]
Any area of the selected flash block 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 Block command to verify all bits are erased.

33.4.11.7 Erase Flash Sector command

The Erase Flash Sector operation erases all addresses in a flash sector.

Table 33-24. Erase Flash Sector Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x09 (ERSSCR)
1	Flash address [23:16] in the flash sector to be erased
2	Flash address [15:8] in the flash sector to be erased
3	Flash address [7:0][1] in the flash sector to be erased

1. Must be 128-bit aligned (Flash address [3:0] = 0000) for program flash, 64-bit aligned (Flash address [2:0] = 000) for data flash.

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 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 33-8).

Table 33-25. 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 128-bit aligned for program flash, 64-bit aligned for data flash	FSTAT[ACCERR]

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 33-25. Erase Flash Sector Command Error Handling (continued)

Error Condition	Error Bit
The selected program flash or data flash sector 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 Section command to verify all bits are erased.

33.4.11.7.1 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.

33.4.11.7.2 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.4.11.7.3 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.

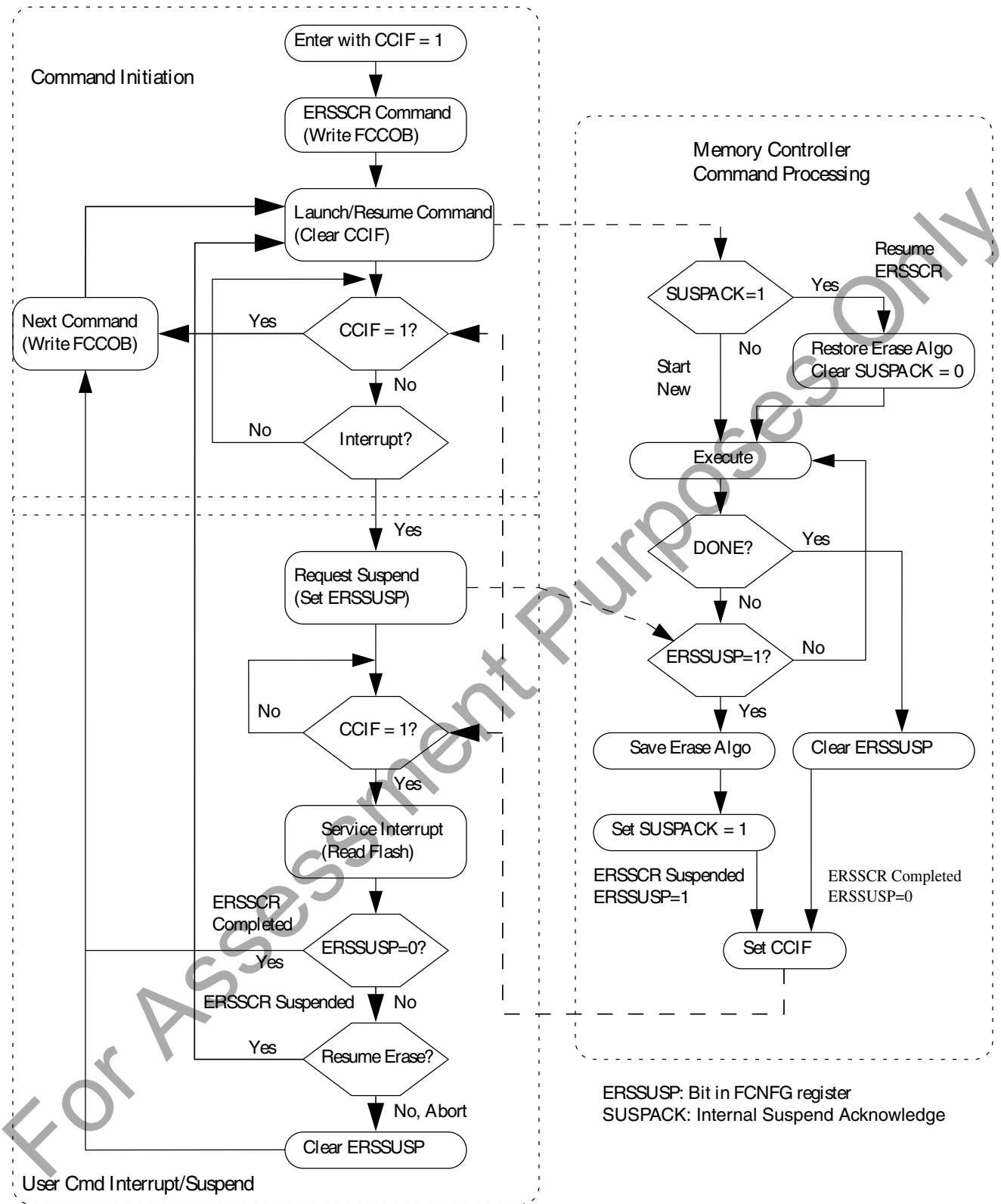
Note

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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required



For Assessment Purposes Only

Figure 33-8. Suspend and Resume of Erase Flash Sector Operation

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.4.11.8 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 0x0000-0x03FF - 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 33-26. 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 double-phrases for program flash, phrases for data flash to program [15:8]
5	Number of double-phrases for program flash, phrases for data flash to program [7:0]

1. Must be 128-bit aligned (Flash address [3:0] = 0000) for program flash, 64-bit aligned (Flash address [2:0] = 000) for data flash.

After clearing CCIF to launch the Program Section command, the FTFE will block access to the FlexRAM 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. Programming, which is not allowed to cross a flash sector boundary, continues until all requested double-phrases for program flash, phrases for data flash 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.

For Assessment Purposes Only

Table 33-27. 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 128-bit aligned for program flash, 64-bit aligned for data flash	FSTAT[ACCERR]
The requested section crosses a program flash sector boundary	FSTAT[ACCERR]
The requested number of double-phrases for program flash, phrases for data flash is zero	FSTAT[ACCERR]
The space required to store data for the requested number of double-phrases for program flash, phrases for data flash is more than one quarter the size of the FlexRAM	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]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

33.4.11.8.1 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 FlexRAM, 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.

For Assessment Purpose

33.4.11.9 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 33-28. 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 33-29,
- 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, 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.

Table 33-29. 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 33-30. 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]

Assessment Purposes

33.4.11.10 Read Once Command

The Read Once command provides read access to a reserved 64-byte field located in the program flash IFR (see Program flash 0 IFR map and Program Once field). Access to the Program Once field is via 8 records, each 8 bytes long. The Program Once field is programmed using the Program Once command described in Program Once command.

Table 33-31. Read Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x41 (RDONCE)
1	Program Once record index (0x00 - 0x07)
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 0x07. During execution of the Read Once command, any attempt to read addresses within the program flash block containing this 64-byte field returns invalid data. The Read Once command can be executed any number of times.

Table 33-32. 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]

33.4.11.11 Program Once command

The Program Once command enables programming to a reserved 64-byte field in the program flash IFR (see Program flash 0 IFR map and Program Once field). Access to the Program Once field is via 8 records, each 8 bytes long. The Program Once field can be

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 IFR cannot be erased.

Table 33-33. Program Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x43 (PGMONCE)
1	Program Once record index (0x00 - 0x07)
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 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 0x07. During execution of the Program Once command, any attempt to read addresses within program flash returns invalid data.

Table 33-34. 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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.4.11.12 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 33-35. 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, 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, 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 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.

Table 33-36. 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.

33.4.11.12.1 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, 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. If the post-erase verify passes, 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

For Assessment Purposes Only

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.

33.4.11.13 Verify Backdoor Access Key command

The Verify Backdoor Access Key command only executes if the mode and security conditions are satisfied (see Flash commands by mode). Execution of the Verify Backdoor Access Key command is further qualified by the FSEC[KEYEN] bits. The Verify Backdoor Access Key command releases security if user-supplied keys in the FCCOB match those stored in the Backdoor Comparison Key bytes of the Flash Configuration Field. The column labeled Flash Configuration Field offset address shows the location of the matching byte in the Flash Configuration Field.

Table 33-37. 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.

For Assessment Purposes Only

Table 33-38. 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]

33.4.11.14 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 33-39. 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, 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, 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. 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.

Table 33-40. 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]

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.4.11.15 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 33-41. 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 Table 2 and EEPROM Data Set Size
2. See Table 3 and FlexNVM partition code

Table 33-42. 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

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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 33-43. Valid FlexNVM Partition Codes

FlexNVM Partition Code DEPART (FCCOB5[3:0])[1]	Data flash Size (Kbytes)	EEPROM-backup Size (Kbytes)
0000	64	0
0011	32	32
0100	0	64
1000	0	64
1010	16	48
1011	32	32
1100	64	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).

Table 33-44. 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 33-42 for valid codes)	FSTAT[ACCERR]
Invalid FlexNVM Partition Code is entered (see Table 33-43 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]

Smart Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

33.4.11.16 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 33-45. Set FlexRAM Function Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x81 (SETRAM)
1	FlexRAM Function Control Code (see Table 33-46)

Table 33-46. 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: • 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

For Assessment Only

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.

Table 33-47. 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]

33.4.12 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.

Table 33-48. FSEC fields

FSEC field	Description
KEYEN	Backdoor Key Access
MEEN	Mass Erase Capability
FSLACC	Factory Security Level Access
SEC	MCU security

33.4.12.1 FTFE Access by Mode and Security

The following table summarizes how access to the FTFE module is affected by security and operating mode.

Table 33-49. FTFE Access Summary

Operating Mode	MCU Security State	
	Unsecure	Secure
NVM Normal	Full command set	
NVM Special	Full command set	Only the Erase All Blocks, Erase All Blocks Unsecure and Read 1s All Blocks commands.

33.4.12.2 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.

33.4.12.2.1 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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

33.4.13 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, and FSEC registers and the FCNFG[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.

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 34
Cyclic Redundancy Check (CRC)

34.1 Introduction

The cyclic redundancy check (CRC) module generates 16/32-bit CRC code for error detection.

The CRC module provides a programmable polynomial and other parameters required to implement a 16-bit or 32-bit CRC standard.

The 16/32-bit code is calculated for 32 bits of data at a time.

34.1.1 Features

Features of the CRC module include:

- Hardware CRC generator circuit using a 16-bit or 32-bit programmable shift register
- Programmable initial seed value and polynomial
- Option for inversion of final CRC result
- 32-bit CPU register programming interface

34.1.2 Block diagram

The following is a block diagram of the CRC.

For Assessment Purposes Only

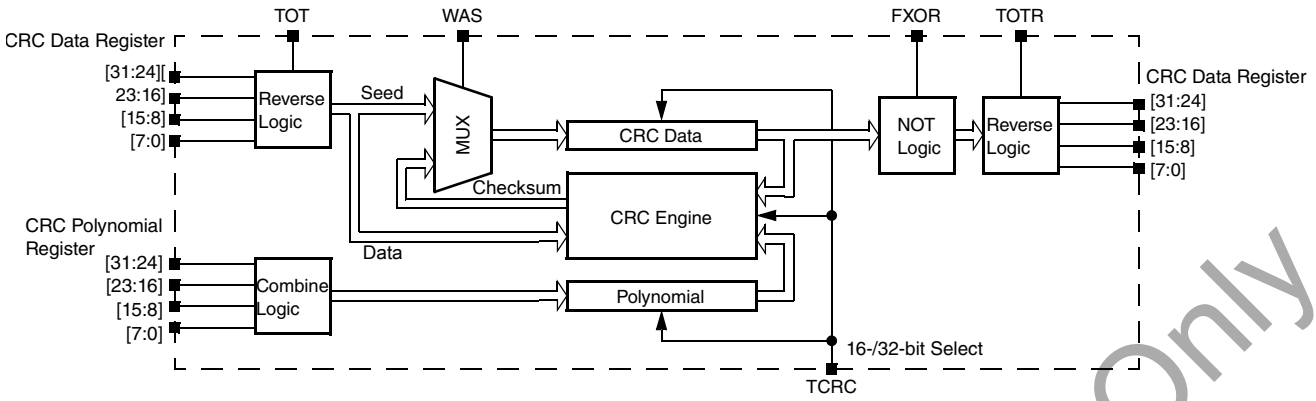


Figure 34-1. Programmable cyclic redundancy check (CRC) block diagram

34.1.3 Modes of operation

Various MCU modes affect the CRC module's functionality.

34.1.3.1 Run mode

This is the basic mode of operation.

34.1.3.2 Low-power modes (Wait or Stop)

Any CRC calculation in progress stops when the MCU enters a low-power mode that disables the module clock. It resumes after the clock is enabled or via the system reset for exiting the low-power mode. Clock gating for this module is dependent on the MCU.

34.2 Memory map and register descriptions

34.2.1 CRC Register Descriptions

For Assessment Purposes Only

34.2.1.1 CRC Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	CRC Data register (DATA)	32	RW	FFFFFFFFh
4h	CRC Polynomial register (GPOLY)	32	RW	00001021h
8h	CRC Control register (CTRL)	32	RW	00000000h

34.2.1.2 CRC Data register (DATA)

34.2.1.2.1 Address

Register	Offset
DATA	0h

34.2.1.2.2 Function

The CRC Data register contains the value of the seed, data, and checksum. When CTRL[WAS] is set, any write to the data register is regarded as the seed value. When CTRL[WAS] is cleared, any write to the data register is regarded as data for general CRC computation.

In 16-bit CRC mode, the HU and HL fields are not used for programming the seed value, and reads of these fields return an indeterminate value. In 32-bit CRC mode, all fields are used for programming the seed value.

When programming data values, the values can be written 8 bits, 16 bits, or 32 bits at a time, provided all bytes are contiguous; with MSB of data value written first.

After all data values are written, the CRC result can be read from this data register. In 16-bit CRC mode, the CRC result is available in the LU and LL fields. In 32-bit CRC mode, all fields contain the result. Reads of this register at any time return the intermediate CRC value, provided the CRC module is configured.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

34.2.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HU								HL							
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	LU								LL							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

34.2.1.2.4 Fields

Field	Function
31-24 HU	CRC High Upper Byte In 16-bit CRC mode (CTRL[TCRC] is 0), this field is not used for programming a seed value. In 32-bit CRC mode (CTRL[TCRC] is 1), values written to this field are part of the seed value when CTRL[WAS] is 1. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation in both 16-bit and 32-bit CRC modes.
23-16 HL	CRC High Lower Byte In 16-bit CRC mode (CTRL[TCRC] is 0), this field is not used for programming a seed value. In 32-bit CRC mode (CTRL[TCRC] is 1), values written to this field are part of the seed value when CTRL[WAS] is 1. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation in both 16-bit and 32-bit CRC modes.
15-8 LU	CRC Low Upper Byte When CTRL[WAS] is 1, values written to this field are part of the seed value. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation.
7-0 LL	CRC Low Lower Byte When CTRL[WAS] is 1, values written to this field are part of the seed value. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation.

34.2.1.3 CRC Polynomial register (GPOLY)

34.2.1.3.1 Address

Register	Offset
GPOLY	4h

For Ass

Freescale Confidential Proprietary - Non-Disclosure Agreement required

34.2.1.3.2 Function

This register contains the value of the polynomial for the CRC calculation. The HIGH field contains the upper 16 bits of the CRC polynomial, which are used only in 32-bit CRC mode. Writes to the HIGH field are ignored in 16-bit CRC mode. The LOW field contains the lower 16 bits of the CRC polynomial, which are used in both 16- and 32-bit CRC modes.

34.2.1.3.3 Diagram

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

34.2.1.3.4 Fields

Field	Function
31-16 HIGH	High Polynominal Half-word Writable and readable in 32-bit CRC mode (CTRL[TCRC] is 1). This field is not writable in 16-bit CRC mode (CTRL[TCRC] is 0).
15-0 LOW	Low Polynominal Half-word Writable and readable in both 32-bit and 16-bit CRC modes.

34.2.1.4 CRC Control register (CTRL)

34.2.1.4.1 Address

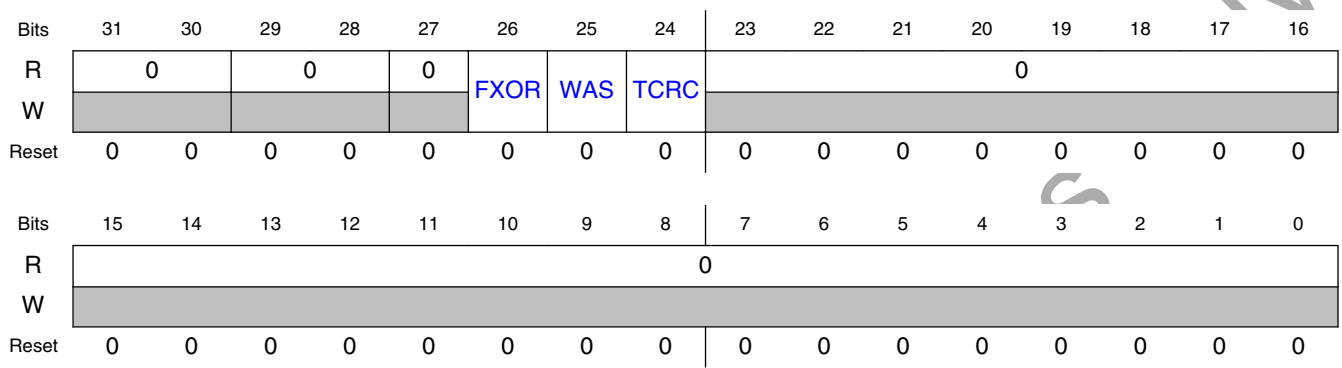
Register	Offset
CTRL	8h

For Assessment Purposes Only

34.2.1.4.2 Function

This register controls the configuration and working of the CRC module. Appropriate bits must be set before starting a new CRC calculation. A new CRC calculation is initialized by asserting CTRL[WAS] and then writing the seed into the CRC data register.

34.2.1.4.3 Diagram



34.2.1.4.4 Fields

Field	Function
31-30 —	Reserved Reserved
29-28 —	Reserved
27 —	Reserved
26 FXOR	Complement Read Of CRC Data Register Some CRC protocols require the final checksum to be XORed with 0xFFFFFFFF or 0xFFFF. Asserting this bit enables on the fly complementing of read data. 0b - No XOR on reading. 1b - Invert or complement the read value of the CRC Data register.
25 WAS	Write CRC Data Register As Seed When asserted, a value written to the CRC data register is considered a seed value. When deasserted, a value written to the CRC data register is taken as data for CRC computation. 0b - Writes to the CRC data register are data values. 1b - Writes to the CRC data register are seed values.
24 TCRC	TCRC Width of CRC protocol. 0b - 16-bit CRC protocol. 1b - 32-bit CRC protocol.
23-0 —	Reserved

Freescale Confidential Proprietary - Non-Disclosure Agreement required

34.3 Functional description

34.3.1 CRC initialization/reinitialization

To enable the CRC calculation, the user must program CRC_CTRL[WAS], CRC_GPOLY,and CRC result inversion in the applicable registers. Asserting CRC_CTRL[WAS] enables the programming of the seed value into the CRC_DATA register.

After a completed CRC calculation, the module can be reinitialized for a new CRC computation by reasserting CRC_CTRL[WAS] and programming a new, or previously used, seed value. All other parameters must be set before programming the seed value and subsequent data values.

34.3.2 CRC calculations

In 16-bit and 32-bit CRC modes, data values can be programmed 8 bits, 16 bits, or 32 bits at a time, provided all bytes are contiguous. Noncontiguous bytes can lead to an incorrect CRC computation.

34.3.2.1 16-bit CRC

To compute a 16-bit CRC:

1. Clear CRC_CTRL[TCRC] to enable 16-bit CRC mode.
2. Write a 16-bit polynomial to the CRC_GPOLY[LOW] field. The CRC_GPOLY[HIGH] field is not usable in 16-bit CRC mode.
3. Set CRC_CTRL[WAS] to program the seed value.
4. Write a 16-bit seed to CRC_DATA[LU:LL]. CRC_DATA[HU:HL] are not used.
5. Clear CRC_CTRL[WAS] to start writing data values.
6. Write data values into CRC_DATA[HU:HL:LU:LL]. A CRC is computed on every data value write, and the intermediate CRC result is stored back into CRC_DATA[LU:LL].
7. When all values have been written, read the final CRC result from CRC_DATA[LU:LL].

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

34.3.2.2 32-bit CRC

To compute a 32-bit CRC:

1. Set CRC_CTRL[TCRC] to enable 32-bit CRC mode.
2. Write a 32-bit polynomial to CRC_GPOLY[HIGH:LOW].
3. Set CRC_CTRL[WAS] to program the seed value.
4. Write a 32-bit seed to CRC_DATA[HU:HL:LU:LL].
5. Clear CRC_CTRL[WAS] to start writing data values.
6. Write data values into CRC_DATA[HU:HL:LU:LL]. A CRC is computed on every data value write, and the intermediate CRC result is stored back into CRC_DATA[HU:HL:LU:LL].
7. When all values have been written, read the final CRC result from CRC_DATA[HU:HL:LU:LL]. The CRC is calculated bytewise, and two clocks are required to complete one CRC calculation.

34.3.3 CRC result complement

When CTRL[FXOR] is set, the checksum is complemented. The CRC result complement function outputs the complement of the checksum value stored in the CRC data register every time the CRC data register is read. When CTRL[FXOR] is cleared, reading the CRC data register accesses the raw checksum value.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 35
Analog-to-Digital Converter (ADC)

35.1 Chip-specific Analog-to-Digital Converter (ADC) information

35.1.1 Instantiation information

This device contains two 12-bit SAR ADC modules.

35.1.1.1 Number of ADC channels

Each SAR ADC supports up to 16 external analog input channels, but the exact ADC channel number present on the device is different with packages as indicated in following table.

For details regarding a specific ADC channel available on a particular package, refer to the signal multiplexing chapter of this MCU.

Table 35-1. ADC external channels per package

ADC Module	100LQFP	64LQFP	100 MAPBGA
ADC0	16	16	16
ADC1	16	11	16

NOTE

Refer to S32K144_IO_Signal_Description_Input_Multiplexing.xlsx attached to this Reference Manual for channel details on specific packages.

For Assessment Purposes Only

35.1.1.2 ADC Connections/Channel Assignment

NOTE

Refer to the S32K144_IO_Signal_Description_Input_Multiplexing.xlsx sheet attached to the Reference Manual for channel assignment details.

35.1.2 DMA Support on ADC

Applications may require continuous sampling of the ADC (4K samples/sec) that may have considerable load on the CPU. Though using PDB to trigger ADC may reduce some CPU load, the ADC supports DMA request functionality for higher performance when the ADC is sampled at a very high rate or cases where PDB is bypassed. The ADC can trigger the DMA (via DMA req) on conversion completion.

For most cases, the DMA request can be directly triggered from ADC conversion completion. The device also support another way to trigger DMA via TRGMUX module. The TRGMUX will provide user a more flexible DMA triggering scheme using software based on different application requirements, for example, the DMA can be triggered after multiple ADC conversion completion instead of every ADC conversion completion.

35.1.3 ADC Hardware Interleaved Channels

On this device, there are several special ADC channels which support hardware interleave between multiple ADCs. Taking ADC0_SE4 and ADC1_SE14 channels as an example, these two channels can work independently, but they can also be hardware interleaved as following diagram. In the hardware interleaved mode, a signal on the pin PTB0 can be sampled by both ADC0 and ADC1. The interleaved mode is enabled by SIM_CHIPCTL[ADC_INTERLEAVE_EN] bits.

The hardware interleave implementation on this device is as following:

- ADC0_SE4 and ADC1_SE14 channels are interleaved on PTB0 pin
- ADC0_SE5 and ADC1_SE15 channels are interleaved on PTB1 pin
- ADC1_SE8 and ADC0_SE8 channels are interleaved on PTB13 pin
- ADC1_SE9 and ADC0_SE9 channels are interleaved on PTB14 pin

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

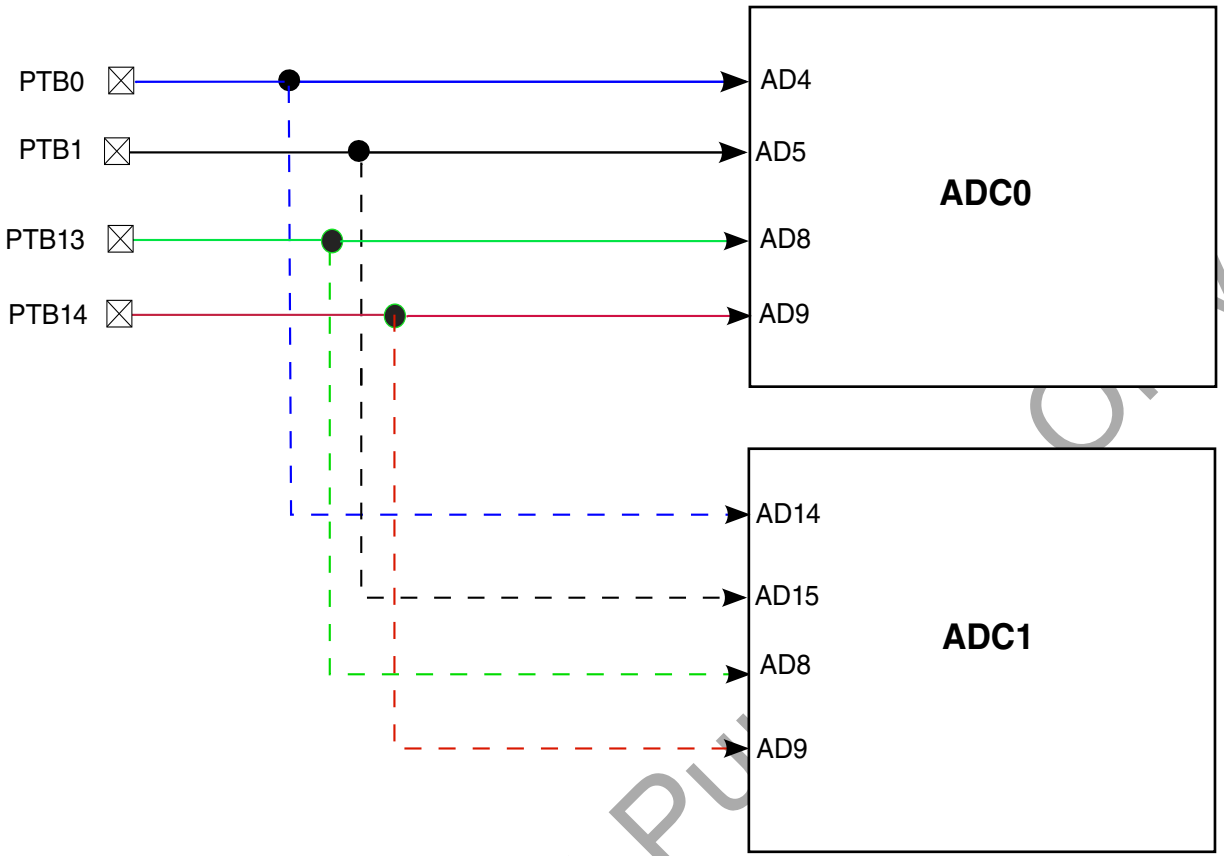


Figure 35-1. ADC0 and ADC1 hardware interleaved channels integration

35.1.4 ADC Reference Options

The ADC supports the following references:

- VREFH/VREFL - connected as the primary reference option
- VALTH/VREFL - connected as an alternate reference option

ADCx_SC2[REFSEL] bit selects the voltage reference sources for ADC. Refer to REFSEL description in ADC chapter for more details.

35.1.5 ADC Trigger Sources

- Triggers are connected through PDB or TRGMUX to provide flexible trigger schemes

For Assessment Purpose Only

- 2×PDB generate triggers and pre-triggers for 2×ADC (the ADC and PDB operate in pairs, as PDB0; ADC0 and PDB1; ADC1), each PDB channel will have up to 8 pre-triggers for ADC channel control, which provides an automatically trigger scheme so that the CPU involvement is not necessary.
- Minimum one external pin per ADC (supported via the TRGMUX)
  - Software must determine relative priority
  - Starts conversion after a single ongoing conversion complete
- CMP out, LPIT, RTC, and LPTMR are capable of triggering each ADC via TRGMUX. See TRGMUX module inter-connectivity in TRGMUX chapter for the modules which are capable of triggering ADC.
- LPIT supports up to 4 pre-triggers, which are limited to be used only on the ADHWTS0 ~ ADHWTS3 of each ADC. For rest of the peripherals, software engagement is required to provide pre-triggers. See SIM_ADCOPT[ADCxSWPRETRG] for configuring the software pre-trigger.

Following specification and diagram are just giving an example to help understanding the ADC trigger scheme. Generally, the ADC support two kind of hardware triggering scheme:

- The default hardware triggering scheme is using PDB to trigger ADC (suggested).
- Another optional hardware triggering scheme is using TRGMUX.
- SIM_ADCOPT[ADCxTRGSEL] bit is used to control the ADC triggering source/ scheme.
  - When ADCxTRGSEL=0, the ADC pre-trigger is coming from PDB directly.
  - When ADCxTRGSEL=1, the ADC pre-trigger is coming from TRGMUX, e.g. LPIT.

NOTE

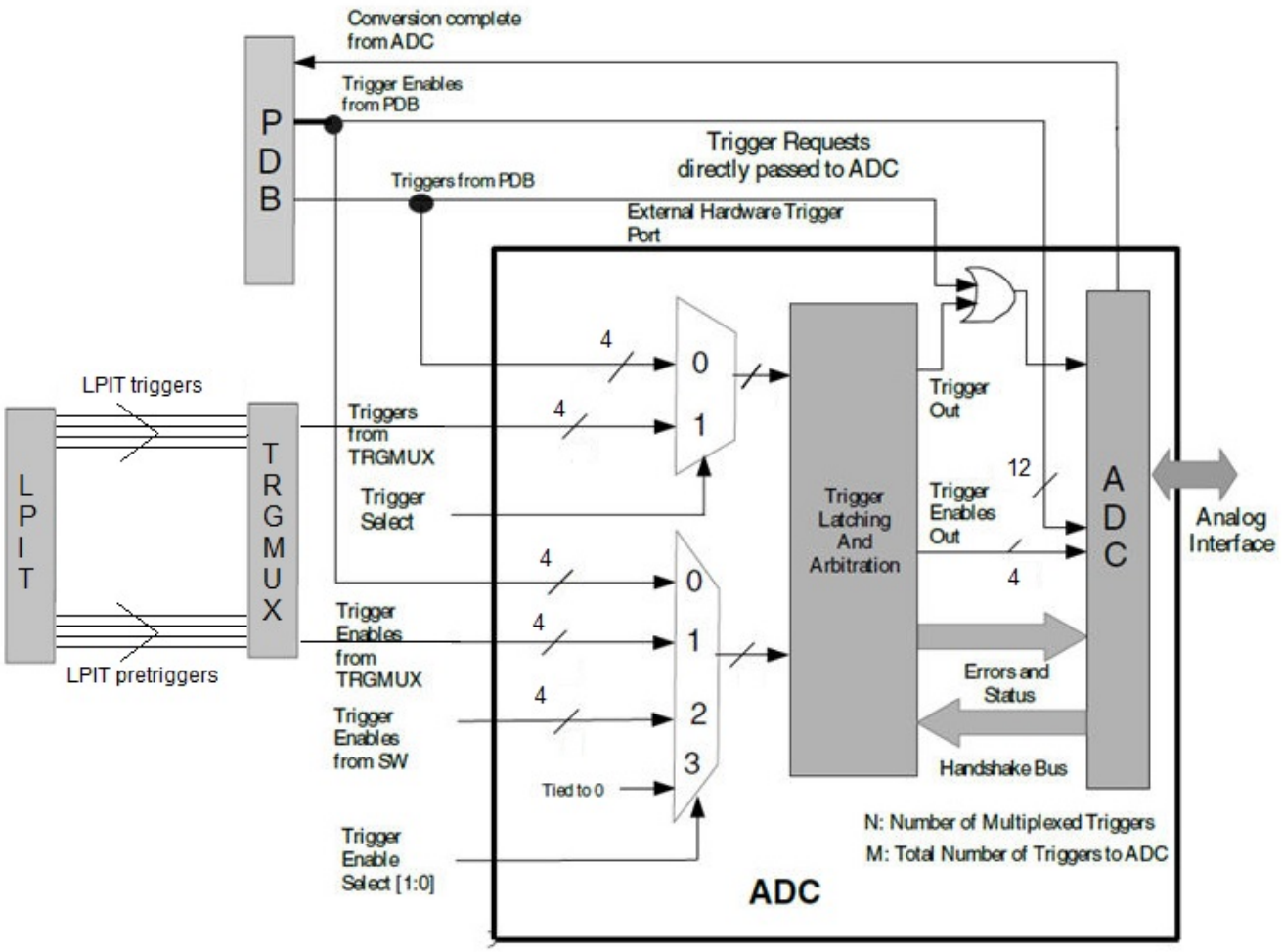
For PDB triggering, each PDB's ADC channel supports up to 16 pre-triggers. But for TRGMUX triggering, only up to 4 pre-triggers are supported.

NOTE

For SIM_ADCOPT[ADC0TRGSEL] or SIM_ADCOPT[ADC1TRGSEL], each PDB supports two ADC channels, and each channel is with 8 pre-triggers. The trigger of two ADC channels are OR'ed together to support up to 16 pre-triggers if necessary.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required



PDB triggering scheme:

PDB triggering scheme is the default and suggested trigger method for ADC. One ADC and one PDB work as one pair, the implementation on this device is: PDB0-ADC0, PDB1-ADC1. The trigger sources for PDB0 and PDB1 can be configured through TRGMUX registers: TRGMUX_PDB0 and TRGMUX_PDB1 respectively. Here we take PDB0-ADC0 as an example to specify the triggering scheme.

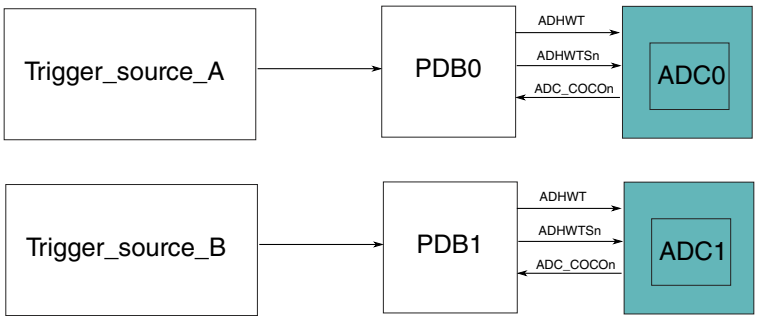
- Set SIM_ADCOPT[ADC*x*TRGSEL]=0. PDB0 channel0 is selected as ADC trigger source.
- PDB0 pre-triggers will connect directly to ADC0 ADHWTS ports to control the channels.
- The ADC0 coco signals are directly feed-backed to PDB0 to deactivate the PDB lock state.

Following are typical case for ADC triggering using PDB:

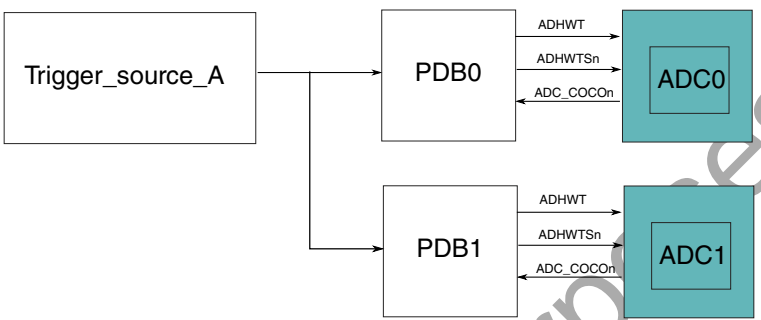
Case 1:

For Assessm

Freescale Confidential Proprietary - Non-Disclosure Agreement required



Case 2:



TRGMUX triggering scheme:

TRGMUX supports many trigger sources, here we take LPIT as an example (typical), but the trigger source can also be others which mentioned above. LPIT supports up to 4 channels, each channel have a trigger and pre-trigger.

- Set SIM_ADCOPT[ADC*x*TRGSEL]=1. TRGMUX out is selected as ADC trigger source.
- Configure TRGMUX to select LPIT triggers as ADC trigger and pre-trigger source.
- TRGMUX only supports up to 4 pre-triggers for each ADC (pre-trigger0 ~ pre-trigger3), the other pre-triggers could not be used with TRGMUX.
- ADC COCO is not required in this case. Software need to take care of the intermission time between each ADC conversion.
- With TRGMUX, a single LPIT could be used to trigger 2 ADCs at same time. This is one of the benefits for TRGMUX triggering, compared with PDB triggering.

NOTE

For other trigger sources other than PDB and LPIT, software engagement is required to provide ADC pre-triggers.

35.1.6 Trigger Selection

Any combination of trigger enable and trigger can be selected; they are independent of each other. But once selected, it cannot be changed on-the-fly. There are defined steps to change the trigger and enable sources.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- Stop the current trigger generation unit (PDB or TRIGMUX) and wait for the triggering status to become idle (poll the status of ADC_SC2[TRGSTLAT] for all 0s). Then change the selection for desired source and then start the trigger generation units.
- If it is required to switch immediately, then stop the current trigger generation unit and flush all the queued triggers of trigger handler block. It will come to idle state immediately or maximum after one conversion time. Poll the status of ADC_SC2[TRGSTLAT] for all 0s. Then change the selection for desired source and then start the trigger generation units. Maximum one conversion may be lost in the second step due to clock domain crossing uncertainty. Uncertainty period of Clearing is equal to 2 cycles (maximum) in ADC operating clock domain (This is due to propagation of ADC_CFG1[CLRLTRG] from host clock to ADC operating clock.) User should wait for maximum X number of cycles in host bus clock; where X = [(2 *period of adc_clk)/period of host bus clock]; before starting the trigger generation unit.

NOTE

If the above restriction is not followed then trigger missed by this process may not be reported.

35.1.7 Trigger Latching and Arbitration

The four lower triggers have the capability to be latched. Irrespective of which source combination is selected, the trigger requests from that source are latched and processed. Latched trigger requests are presented to ADC one at a time. The next request is given only when the processing of current trigger request is completed by ADC. So at a time only one request is under processing and all other requests are latched. If a trigger request appears again on any of the 0-3enables/triggers, which is either under process by ADC or already latched in trigger handler the new request is ignored and an error is signaled for that trigger in register ADC_SC2[TRGSTERR].

The relation among the trigger requests from the selected source can be in random, one-hot or simultaneous. But they will be serviced in round-robin basis. For example, after P th request is processed [0 < P <(N-1)], then it will start searching from (P+1) followed by (P+2) till it finds the next latched request; this searching rolls over to 0 after (N-1) th trigger and continues till P.

To work with ADC having multiplexed trigger user needs to follow the below sequences of programming for different cases:

- Case 1: Initialization / start of conversion:
  a. Initialize / re-program(in case of intermediate start) SAR ADC for desired operation (see ADC section for details).

For Assessment Purposes Only

b. Select the trigger source for Trigger handler block (configuration for this resides in a separate module on device, see device specific section).
c. Configure and start the trigger generation module (PDB or TRGMUX).

- Case 2: Changing trigger source / Multiplexer control:
  a. Stop the trigger generation module (PDB or TRGMUX).
  b. Do any of the following:

     Option 1: Wait for the latched triggers to be processed and trigger handler to be idle (poll the status of ADC_SC2[TRGSTLAT] for all 0s).

     Option 2:
     1. Flush the latched trigger requests in trigger handler block (Write a 1b to the ADC_CFG1[CLRLTRG]).
     2. Wait for the idle status of trigger handler (poll the status of ADC_SC2[TRGSTLAT] for all 0).

  c. Re-select the trigger source for Trigger handler block (configuration for this resides in a separate module on device, see device specific section).
  d. Configure and start the trigger generation module (PDB or TRGMUX).
- Case 3: Stopping of conversion
  a. Stop the trigger generation module (PDB or TRGMUX).
  b. Wait for the latched triggers to be processed and trigger handler to be idle (poll the status of ADC_SC2[TRGSTLAT] for all 0).
  c. Stop the SAR ADC and clear interrupts after required processing (see ADC section for details).
- Case 4: Changing trigger source / Multiplexer control in case of ongoing continuous conversion
  a. Stop the trigger generation module (PDB or TRGMUX).
  b. Read any configuration register of ADC
  c. Write the read value from above in the same register.
  d. This write will abort the ongoing continuous conversion.
  e. Re-select the trigger source for Trigger handler block (configuration for this resides in a separate module on device, see device specific section).
  f. Configure and start the trigger generation module (PDB or TRIGMUX).
- Case 5: Clearing the error status (applicable for TRGMUX case only)
  a. Read the error status (ADC_SC2[TRGSTERR]).
  b. If any of the above register bit is 1, then stop the trigger generation module (TRGMUX).
  c. Wait for the idle status of trigger handler (poll the status of ADC_SC2[TRGSTLAT] for all 0).
  d. Clear the above register bit by writing 1 in it.
  e. Wait for the latched triggers to be processed and trigger handler to be idle (poll the status of ADC_SC2[TRGSTLAT] for all 0).

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

f. Re-select the trigger source for Trigger handler block (configuration for this resides in a separate module on device, see device specific section).
g. Configure and start the trigger generation module (PDB or TRGMUX).

NOTE

Latched status clearing by writing 1 to ADC_CFG1[CLRLTRG] should not be done while PDB triggering through trigger handler as PDB works in a closed loop with COCO flag from ADC and provides one trigger at a time. Clearing the latched status while it is under processing within trigger handler (before being launched to ADC) will stop its future conversion and there will be no COCO flag. In such case, PDB might enter an indeterminate state.

35.1.8 ADC low-power modes

The ADC will be available in STOP, VLPR, VLPW, and VLPS mode.

NOTE

When in VLPx mode, the ADC clock source is only limited to OSC and SIRC.

35.1.9 ADC Trigger Concept – Use Case

FTM module support counter init trigger and channel match trigger, these triggers could be used as trigger input of PDB, PDB then be used to trigger other modules like ADC. Each ADC channel in PDB module supports up to 8 pre-triggers, which could be used as ADC hardware channel selection to precondition the ADC block prior to actual trigger. The ADC trigger is initiated after pre-trigger to trigger ADC conversion. The waveforms shown in following diagram illustrate the pre-trigger and trigger output of PDB to ADC. Every time when one PDB pre-trigger and trigger output starts an ADC conversion, an internal lock associated with the corresponding pre-trigger is activated. This lock becomes inactive when receiving COCO signal from ADC.

For Assessment Purposes Only

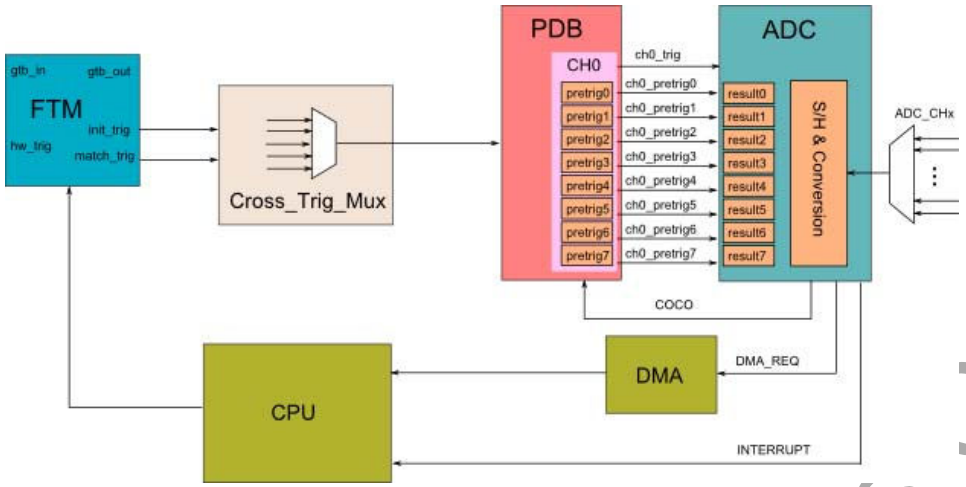


Figure 35-2. PWM Load Diagnosis – ADC Trigger Concept (block diagram)

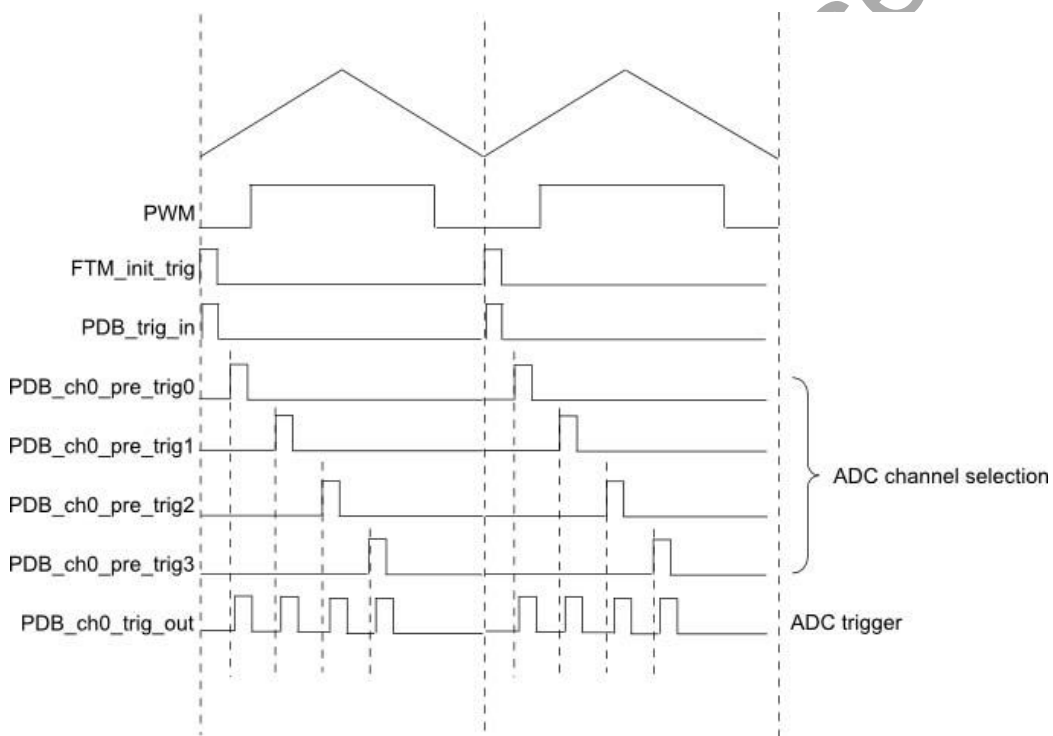


Figure 35-3. PWM Load Diagnosis – ADC Trigger Concept 1 (Timing)

For ASIC Samples Only

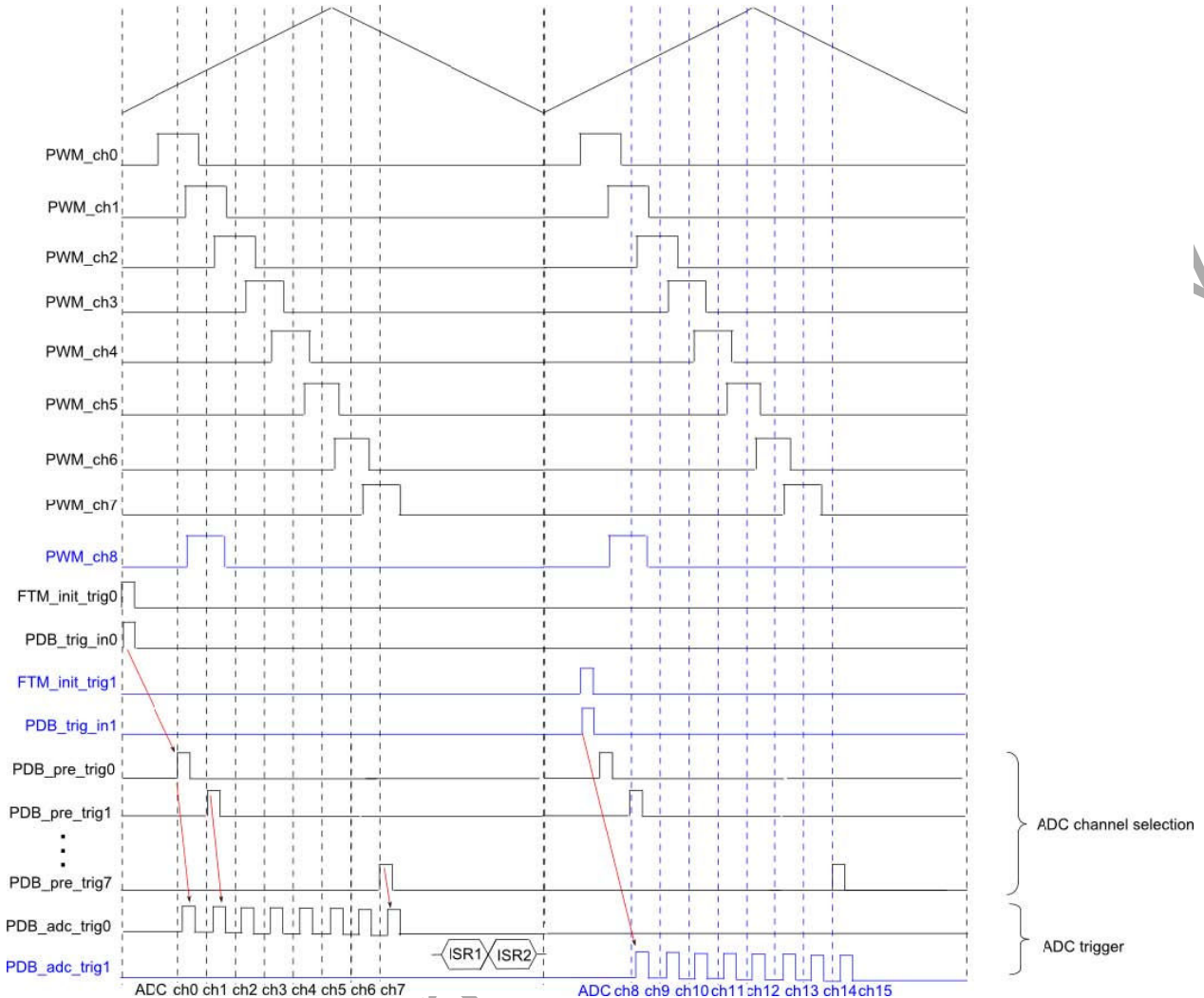


Figure 35-4. PWM Load Diagnosis – ADC Trigger Concept 2 (Timing)

35.1.10 ADC self-test and calibration scheme

On Power Up (following a POR), the ADC self-test and calibration will be executed after the flash has been made available. Subsequent exits from power modes such as VLPS the self-test and calibration will not execute automatically. However, there will be a user option to enable the self-test / calibration in the various run modes at any time.

35.2 Introduction

The 12-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

NOTE

For the chip specific modes of operation, see the power management information of the device.

35.2.1 Features

Following are the features of the ADC module.

- Linear successive approximation algorithm with up to 12-bit resolution
- Up to 16 single-ended external analog inputs
- Output modes:
  - single-ended 12-bit, 10-bit, and 8-bit 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
- 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

For Assessment Purposes Only

35.2.2 Block diagram

The following figure is the ADC module block diagram.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

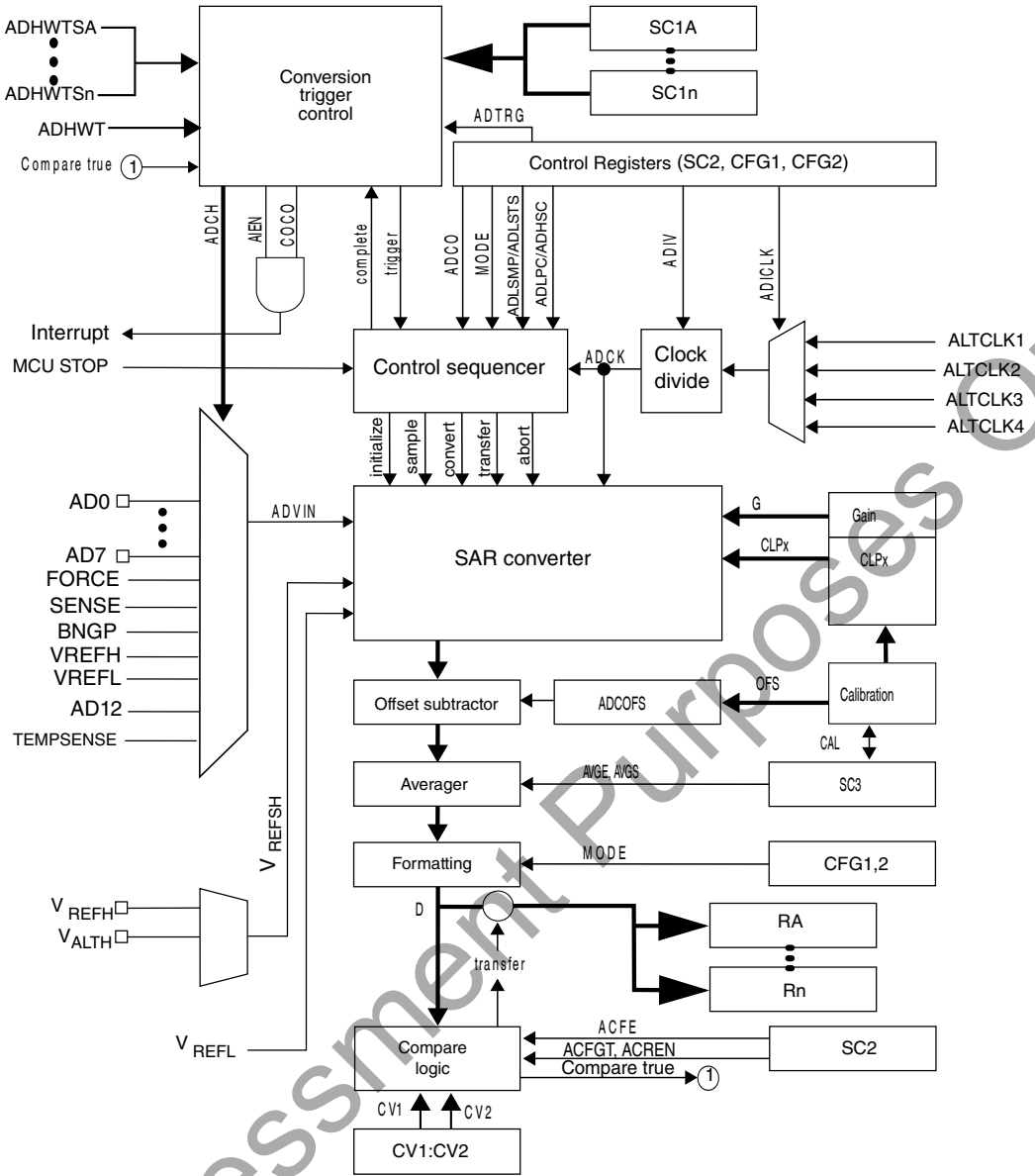


Figure 35-5. ADC block diagram

35.3 ADC signal descriptions

The ADC module supports up to 16 single-ended inputs.

The ADC also requires four supply/reference/ground connections.

NOTE

For the number of channels supported on this device, see the chip-specific ADC information.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 35-2. ADC signal descriptions

Signal	Description	I/O
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

35.3.1 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.

35.3.2 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}.

35.3.3 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 the voltage reference pairs for V_{REFSH} and V_{REFSL} by configuring V_{REFSH} as V_{REFH} or $_{VALTH}$. 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}) alternate (V_{ALTLH} and V_{REFL}). These voltage references are selected using SC2[REFSEL]. The alternate voltage reference, V_{ALTH} may select additional external pin or internal source depending on MCU configuration. See the chip configuration information on the Voltage References specific to this MCU.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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}.

35.3.4 Analog Channel Inputs (ADx)

The ADC module supports up to 8 analog inputs. A analog input is selected for conversion through the SC1[ADCH] channel select bits.

35.4 Memory map and register definitions

This section describes the ADC registers.

ADC memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	ADC Status and Control Registers 1 (ADC_SC1A)	32	R/W	0000_001Fh	35.4.1/695
4	ADC Status and Control Registers 1 (ADC_SC1B)	32	R/W	0000_001Fh	35.4.1/695
8	ADC Status and Control Registers 1 (ADC_SC1C)	32	R/W	0000_001Fh	35.4.1/695
C	ADC Status and Control Registers 1 (ADC_SC1D)	32	R/W	0000_001Fh	35.4.1/695
10	ADC Status and Control Registers 1 (ADC_SC1E)	32	R/W	0000_001Fh	35.4.1/695
14	ADC Status and Control Registers 1 (ADC_SC1F)	32	R/W	0000_001Fh	35.4.1/695
18	ADC Status and Control Registers 1 (ADC_SC1G)	32	R/W	0000_001Fh	35.4.1/695
1C	ADC Status and Control Registers 1 (ADC_SC1H)	32	R/W	0000_001Fh	35.4.1/695
20	ADC Status and Control Registers 1 (ADC_SC1I)	32	R/W	0000_001Fh	35.4.1/695
24	ADC Status and Control Registers 1 (ADC_SC1J)	32	R/W	0000_001Fh	35.4.1/695
28	ADC Status and Control Registers 1 (ADC_SC1K)	32	R/W	0000_001Fh	35.4.1/695
2C	ADC Status and Control Registers 1 (ADC_SC1L)	32	R/W	0000_001Fh	35.4.1/695
30	ADC Status and Control Registers 1 (ADC_SC1M)	32	R/W	0000_001Fh	35.4.1/695
34	ADC Status and Control Registers 1 (ADC_SC1N)	32	R/W	0000_001Fh	35.4.1/695
38	ADC Status and Control Registers 1 (ADC_SC1O)	32	R/W	0000_001Fh	35.4.1/695
3C	ADC Status and Control Registers 1 (ADC_SC1P)	32	R/W	0000_001Fh	35.4.1/695
40	ADC Configuration Register 1 (ADC_CFG1)	32	R/W	0000_0000h	35.4.2/698
44	ADC Configuration Register 2 (ADC_CFG2)	32	R/W	0000_000Ch	35.4.3/699
48	ADC Data Result Register (ADC_RA)	32	R	0000_0000h	35.4.4/699
4C	ADC Data Result Register (ADC_RB)	32	R	0000_0000h	35.4.4/699
50	ADC Data Result Register (ADC_RC)	32	R	0000_0000h	35.4.4/699

Table continues on the next page...

Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ADC memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
54	ADC Data Result Register (ADC_RD)	32	R	0000_0000h	35.4.4/699
58	ADC Data Result Register (ADC_RE)	32	R	0000_0000h	35.4.4/699
5C	ADC Data Result Register (ADC_RF)	32	R	0000_0000h	35.4.4/699
60	ADC Data Result Register (ADC_RG)	32	R	0000_0000h	35.4.4/699
64	ADC Data Result Register (ADC_RH)	32	R	0000_0000h	35.4.4/699
68	ADC Data Result Register (ADC_RI)	32	R	0000_0000h	35.4.4/699
6C	ADC Data Result Register (ADC_RJ)	32	R	0000_0000h	35.4.4/699
70	ADC Data Result Register (ADC_RK)	32	R	0000_0000h	35.4.4/699
74	ADC Data Result Register (ADC_RL)	32	R	0000_0000h	35.4.4/699
78	ADC Data Result Register (ADC_RM)	32	R	0000_0000h	35.4.4/699
7C	ADC Data Result Register (ADC_RN)	32	R	0000_0000h	35.4.4/699
80	ADC Data Result Register (ADC_RO)	32	R	0000_0000h	35.4.4/699
84	ADC Data Result Register (ADC_RP)	32	R	0000_0000h	35.4.4/699
88	Compare Value Registers (ADC_CV1)	32	R/W	0000_0000h	35.4.5/700
8C	Compare Value Registers (ADC_CV2)	32	R/W	0000_0000h	35.4.5/700
90	Status and Control Register 2 (ADC_SC2)	32	R/W	0000_0000h	35.4.6/701
94	Status and Control Register 3 (ADC_SC3)	32	R/W	0000_0000h	35.4.7/703
98	BASE Offset Register (ADC_BASE_OFS)	32	R/W	0000_0040h	35.4.8/705
9C	ADC Offset Correction Register (ADC_OFS)	32	R/W	0000_0000h	35.4.9/705
A0	USER Offset Correction Register (ADC_USR_OFS)	32	R/W	0000_0000h	35.4.10/ 706
A4	ADC X Offset Correction Register (ADC_XOFS)	32	R/W	0000_0030h	35.4.11/ 706
A8	ADC Y Offset Correction Register (ADC_YOFS)	32	R/W	0000_0037h	35.4.12/ 707
AC	ADC Gain Register (ADC_G)	32	R/W	0000_02F0h	35.4.13/ 707
B0	ADC User Gain Register (ADC_UG)	32	R/W	0000_0004h	35.4.14/ 708
B4	ADC General Calibration Value Register (ADC_CLPS)	32	R/W	See section	35.4.15/ 708
B8	ADC Plus-Side General Calibration Value Register (ADC_CLP3)	32	R/W	See section	35.4.16/ 709
BC	ADC Plus-Side General Calibration Value Register (ADC_CLP2)	32	R/W	See section	35.4.17/ 709
C0	ADC Plus-Side General Calibration Value Register (ADC_CLP1)	32	R/W	See section	35.4.18/ 710
C4	ADC Plus-Side General Calibration Value Register (ADC_CLP0)	32	R/W	See section	35.4.19/ 710
C8	ADC Plus-Side General Calibration Value Register (ADC_CLPX)	32	R/W	See section	35.4.20/ 711

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ADC memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
CC	ADC Plus-Side General Calibration Value Register (ADC_CLP9)	32	R/W	See section	35.4.21/ 712
D0	ADC General Calibration Value Register (ADC_CLPS_OFS)	32	R/W	0000_0000h	35.4.22/ 712
D4	ADC Plus-Side General Calibration Value Register (ADC_CLP3_OFS)	32	R/W	0000_0000h	35.4.23/ 713
D8	ADC Plus-Side General Calibration Value Register (ADC_CLP2_OFS)	32	R/W	0000_0000h	35.4.24/ 713
DC	ADC Plus-Side General Calibration Value Register (ADC_CLP1_OFS)	32	R/W	0000_0000h	35.4.25/ 714
E0	ADC Plus-Side General Calibration Value Register (ADC_CLP0_OFS)	32	R/W	0000_0000h	35.4.26/ 714
E4	ADC Plus-Side General Calibration Value Register (ADC_CLPX_OFS)	32	R/W	0000_0440h	35.4.27/ 715
E8	ADC Plus-Side General Calibration Value Register (ADC_CLP9_OFS)	32	R/W	0000_0240h	35.4.28/ 715

35.4.1 ADC Status and Control Registers 1 (ADC_SC1*n*)

SC1A is used for both software and hardware trigger modes of 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.

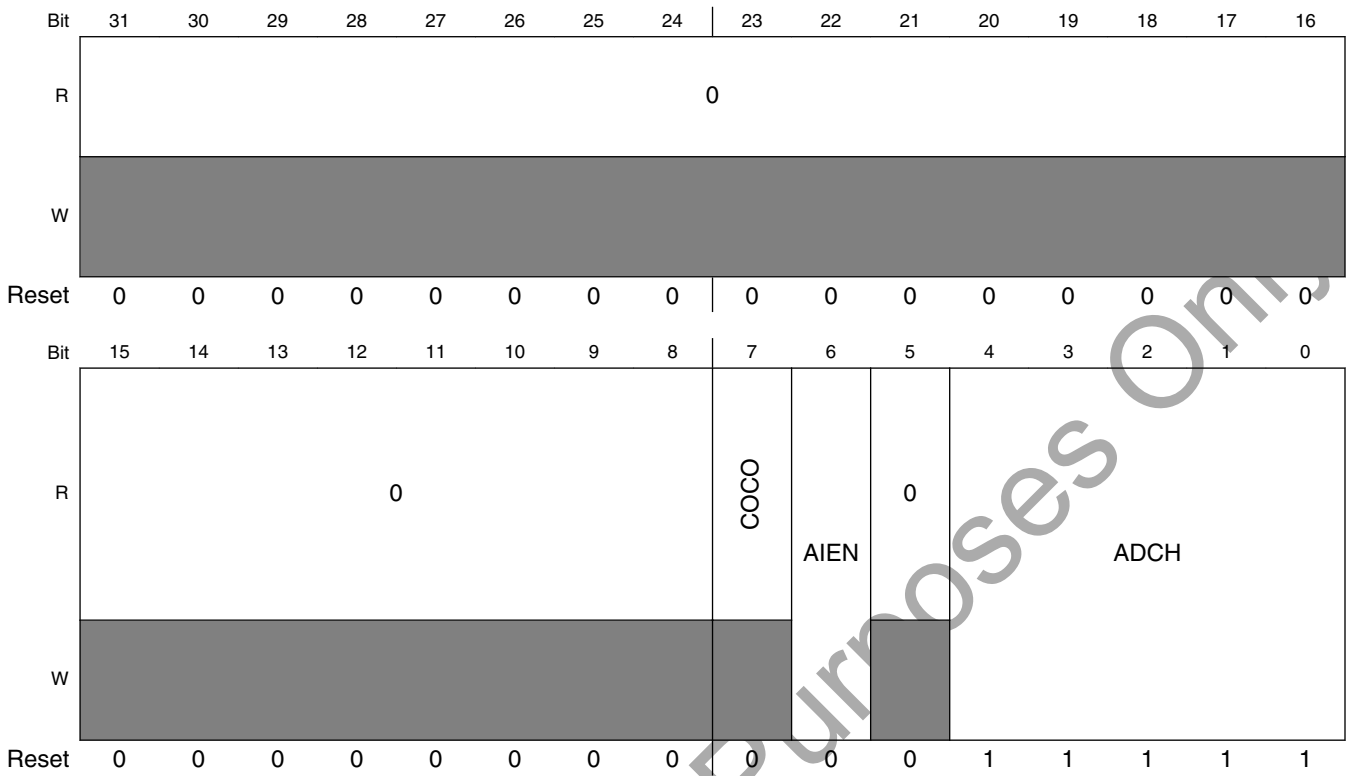
Writing SC1A while SC1A is actively controlling a conversion aborts the current conversion. In Software Trigger mode, when SC2[ADTRG]=0, writes to SC1A subsequently initiate a new conversion, if SC1A[ADCH] contains a value other than all 1s.

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.

For Assessment Purp

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 0h offset + (4d × i), where i=0d to 15d



Purposes Only

ADC_SC1*n* 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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ADCH	Input channel select

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ADC_SC1*n* field descriptions (continued)

Field	Description
	Selects one of the input channels. **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 bits are all set (i.e. ADCH set to all 1s). 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 AD0 is selected as input. 00001 AD1 is selected as input. 00010 AD2 is selected as input. 00011 AD3 is selected as input. 00100 AD4 is selected as input. 00101 AD5 is selected as input. 00110 AD6 is selected as input. 00111 AD7 is selected as input. 01000 AD8 is selected as input. 01001 AD9 is selected as input. 01010 AD10 is selected as input. 01011 AD11 is selected as input. 01100 AD12 is selected as input. 01101 AD13 is selected as input. 01110 AD14 is selected as input. 01111 AD15 is selected as input. 10000 Reserved. 10001 Reserved. 10010 AD18 is selected as input. 10011 AD19 is selected as input. 10100 Reserved. 10101 AD21 is selected as input. 10110 AD22 is selected as input. 10111 AD23 is selected as input. 11000 Reserved 11001 Reserved 11010 Temp Sensor 11011 Band Gap 11100 AD28 is selected as input. 11101 V_{REFSH} is selected as input. Voltage reference selected is determined by SC2[REFSEL]. 11110 V_{REFSL} is selected as input. Voltage reference selected is determined by SC2[REFSEL]. 11111 Module is disabled..

Freescale Confidential Proprietary - Non-Disclosure Agreement required

35.4.2 ADC Configuration Register 1 (ADC_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: 0h base + 40h offset = 40h

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	ADIV		0	MODE		ADICLK	
W								1								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADC_CFG1 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 CLRLTRG	Clear Latch Trigger in Trigger Handler Block Writing a 1'b1 to this bit clears all the latched triggers inside trigger handler except one under processing. Writing 1'b0 has no effect. This is write-only one bit and self-clearing immediately.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–2 MODE	Conversion mode selection Selects the ADC resolution mode. 00 8-bit conversion. 01 12-bit conversion. 10 10-bit conversion. 11 Reserved
ADICLK	Input Clock Select Selects the input clock source to generate the internal clock, ADCK. 00 Alternate clock 1 (ADC_ALTCLK1)

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ADC_CFG1 field descriptions (continued)

Field	Description
	01 Alternate clock 2 (ADC_ALTCLK2) 10 Alternate clock 3 (ADC_ALTCLK3) 11 Alternate clock 4 (ADC_ALTCLK4)

35.4.3 ADC Configuration Register 2 (ADC_CFG2)

Configuration Register 2 (CFG2) selects the long sample time duration during long sample mode.

NOTE

Writing 0 is not supported on this register.

Address: 0h base + 44h offset = 44h

Bit	31	30	29	28	27	26	25	24	23	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																								SMPLTS							
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	1	0	0

ADC_CFG2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SMPLTS	Sample Time Select Selects a sample time of 2 to 256 ADCK clock cycles. The value written to this register is the desired sample time minus 1. A sample time of 1 is not supported. 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.

35.4.4 ADC Data Result Register (ADC_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.

For Assessment Purposes Only

The following table describes the behavior of the data result registers in the different modes of operation.

Table 35-3. Data result register description

Conversion mode	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
12-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
10-bit single-ended	0	0	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
8-bit single-ended	0	0	0	0	D	D	D	D	D	D	D	D	Unsigned right-justified

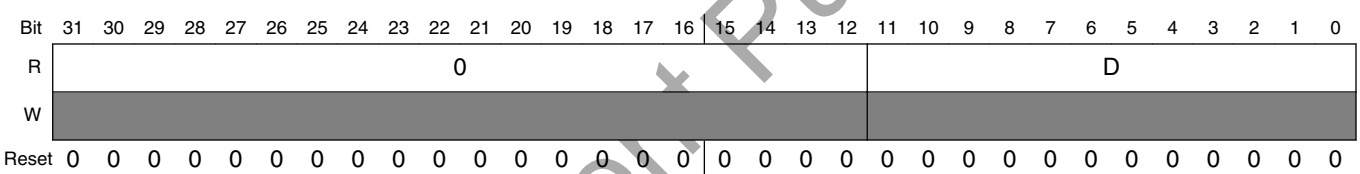
NOTE

D: Data

NOTE

The data result registers are read-only. Writing to these registers generates transfer error.

Address: 0h base + 48h offset + (4d × i), where i=0d to 15d



ADC_R*n* field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
D	Data result

35.4.5 Compare Value Registers (ADC_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. 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.

For Assessment Purposes Only

Address: 0h base + 88h offset + (4d × i), where i=0d to 1d

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

ADC_CV*n* field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CV	Compare Value.

35.4.6 Status and Control Register 2 (ADC_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: 0h base + 90h offset = 90h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				TRGSTERR				0				TRGSTLAT			
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	TRGPRNUM			0				ADACT	ADTRG	ACFE	ACFGT	ACREN	DMAEN	REFSEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

For Assessment Purposes Only

ADC_SC2 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 TRGSTERR	Error in Multiplexed Trigger Request Error Signal to indicate if multiplexed hardware trigger request from source is missed; it is either already latched or under process. To be cleared by writing 1 to individual bits. Individual bit corresponds to individual trigger request. Bit 24 corresponds to 0th trigger request, bit 25 corresponds to 1st trigger request and so on.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 TRGSTLAT	Trigger Status Status to indicate multiplexed hardware trigger requests from source are latched. Individual bit corresponds to individual trigger request. i.e. Bit 16 corresponnds to 0th trigger request, bit 17 corresponds to 1st trigger request and so on.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–13 TRGPRNUM	Trigger Process Number Shows trigger number under process. This has to be qualified with a 1b value for the corresponding trigger latch status.
12–8 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: • 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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ADC_SC2 field descriptions (continued)

Field	Description
	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 voltage, that is, V_{ALTH}. This voltage may be additional external pin or internal source depending on the MCU configuration. See the chip configuration information for details specific to this MCU. 10 Reserved 11 Reserved

For Assessment

35.4.7 Status and Control Register 3 (ADC_SC3)

The Status and Control Register 3 (SC3) controls the calibration, continuous convert, and hardware averaging functions of the ADC module.

Address: 0h base + 94h offset = 94h

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

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ADC_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. Once started, the calibration routine cannot be interrupted by writes to the ADC registers or the results will be invalid . Setting CAL will abort any current conversion. **NOTE:** For calibration, it is mandatory to use averaging and average number 32. **NOTE:** If several ADCs are on a device, they should be calibrated sequentially. No parallel calibrations of ADCs are allowed because they will disturb each other.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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.

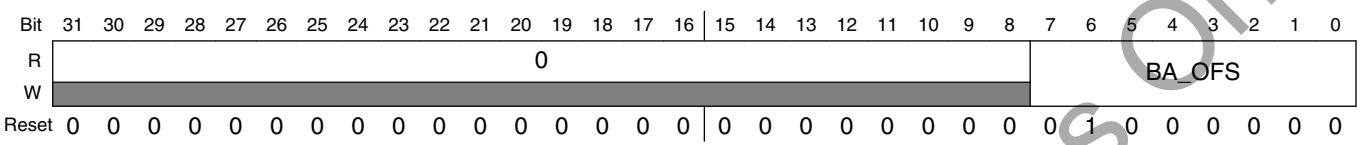
For R

Freescale Confidential Proprietary - Non-Disclosure Agreement required

35.4.8 BASE Offset Register (ADC_BASE_OFS)

The BASE Offset Register (BASE_OFS) contains the offset value used by the calibration algorithm to determine Offset Calibration Value (OFS).

Address: 0h base + 98h offset = 98h



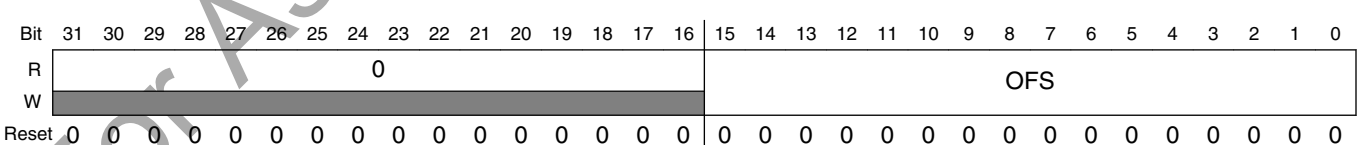
ADC_BASE_OFS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
BA_OFS	Base Offset Error Correction Value

35.4.9 ADC Offset Correction Register (ADC_OFS)

The ADC Offset Correction Register (OFS) contains the calibration-generated offset error correction value (OFS). The value in (BA_OFF) is used in the calibration algorithm to calculate the offset correction value that gets stored in the (OFS) register. 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.

Address: 0h base + 9Ch offset = 9Ch



ADC_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

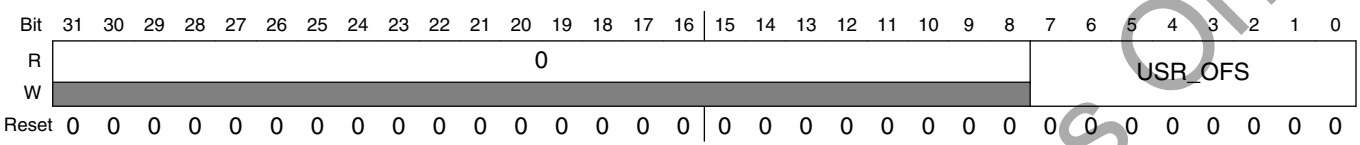
For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

35.4.10 USER Offset Correction Register (ADC_USR_OFS)

The USER Offset Correction Register (USR_OFS) contains the user defined offset error correction value used in the conversion result error correction algorithm.

Address: 0h base + A0h offset = A0h



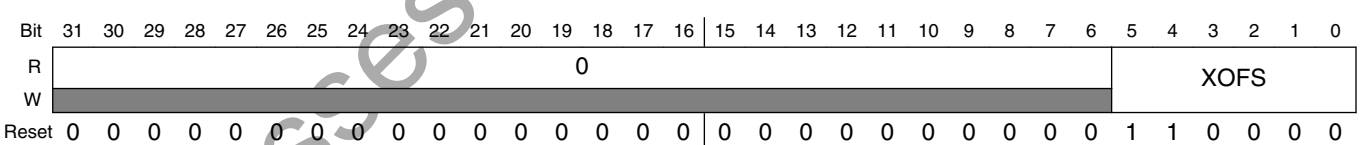
ADC_USR_OFS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
USR_OFS	USER Offset Error Correction Value

35.4.11 ADC X Offset Correction Register (ADC_XOFS)

The ADC X Offset Correction Register (XOFS) contains the X offset for offset error correction used in the conversion result error correction algorithm.

Address: 0h base + A4h offset = A4h



ADC_XOFS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
XOFS	X Offset Error Correction Value

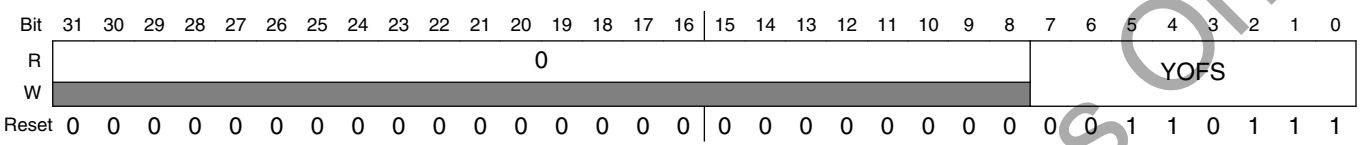
Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

35.4.12 ADC Y Offset Correction Register (ADC_YOFS)

The ADC Y Offset Correction Register (YOFS) contains the offset for error correction used in the conversion result error correction algorithm.

Address: 0h base + A8h offset = A8h



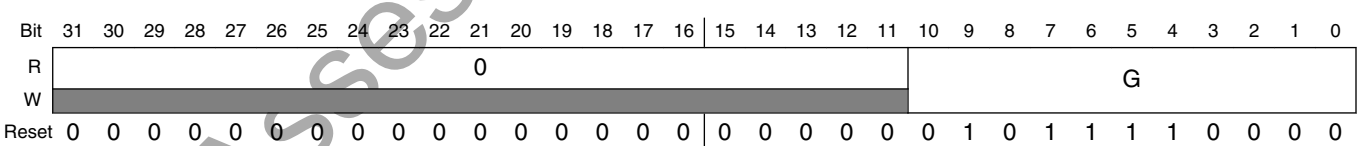
ADC_YOFS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
YOFS	Y Offset Error Correction Value

35.4.13 ADC Gain Register (ADC_G)

The Gain Register (G) contains the gain error correction for the overall conversion. G, a 12-bit real number in binary format, is the gain adjustment factor. This register value is determined and uploaded by the calibration algorithm.

Address: 0h base + ACh offset = ACh



ADC_G field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
G	Gain

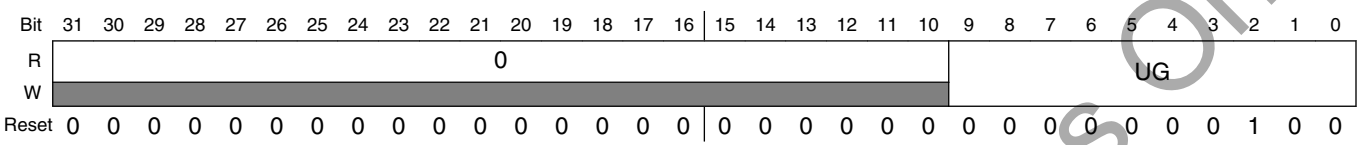
Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

35.4.14 ADC User Gain Register (ADC_UG)

The User Gain Register (UG) contains the user gain error correction. Allow user to adjust the final calibration gain value.

Address: 0h base + B0h offset = B0h



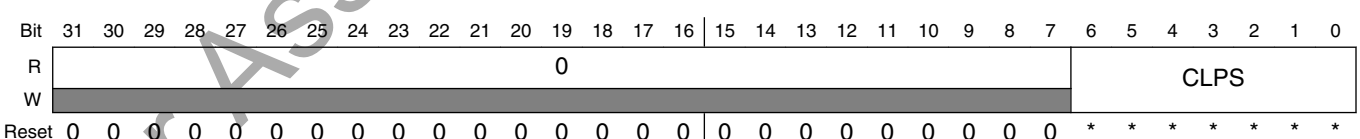
ADC_UG field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
UG	User Gain

35.4.15 ADC General Calibration Value Register (ADC_CLPS)

The 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[7:0], CLP1[8:0], CLP2[9:0], CLP3[9:0], CLPS[6:0], CLPX[6:0]and CLP9[6:0]. 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.

Address: 0h base + B4h offset = B4h



* Notes:
- CLPS field: Reset values are loaded out of IFR.

ADC_CLPS field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

ADC_CLPS field descriptions (continued)

Field	Description
CLPS	Calibration Value Calibration Value

35.4.16 ADC Plus-Side General Calibration Value Register (ADC_CLP3)

For more information, see CLP3 register description.

Address: 0h base + B8h offset = B8h

Bit	31	30	29	28	27	26	25	24	23	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	*	*	*	*	*	*	*	*	*	*

* Notes:
- CLP3 field: Reset values are loaded out of IFR.

ADC_CLP3 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP3	Calibration Value Calibration Value

35.4.17 ADC Plus-Side General Calibration Value Register (ADC_CLP2)

For more information, see CLP2 register description.

Address: 0h base + BCh offset = BCh

Bit	31	30	29	28	27	26	25	24	23	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	*	*	*	*	*	*	*	*	*	*

* Notes:
- CLP2 field: Reset values are loaded out of IFR.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

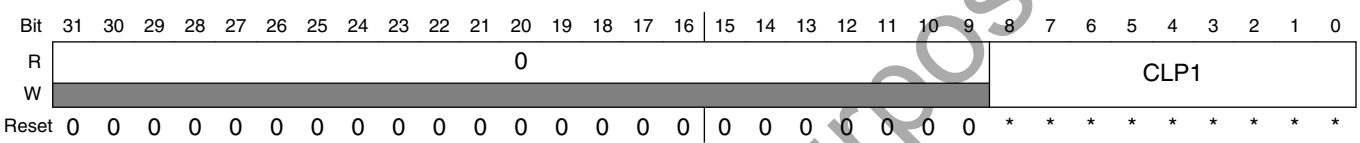
ADC_CLP2 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP2	Calibration Value Calibration Value

35.4.18 ADC Plus-Side General Calibration Value Register (ADC_CLP1)

For more information, see CLPD register description.

Address: 0h base + C0h offset = C0h



* Notes:
- CLP1 field: Reset values are loaded out of IFR.

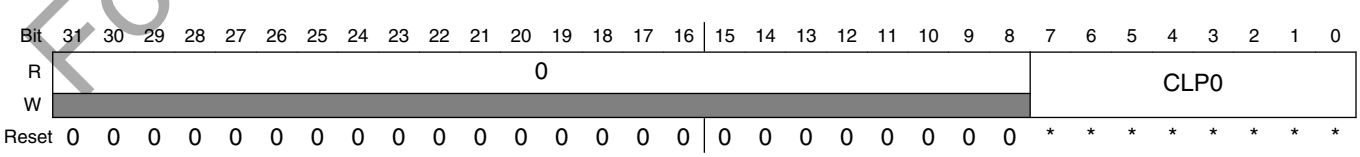
ADC_CLP1 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP1	Calibration Value Calibration Value

35.4.19 ADC Plus-Side General Calibration Value Register (ADC_CLP0)

For more information, see CLPD register description.

Address: 0h base + C4h offset = C4h



* Notes:
- CLP0 field: Reset values are loaded out of IFR.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

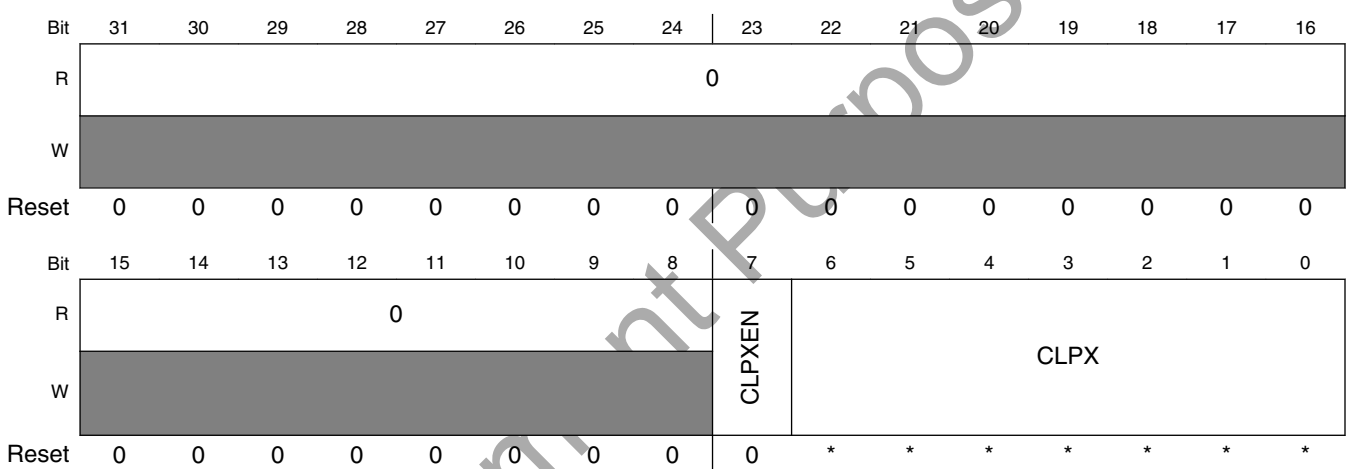
ADC_CLP0 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP0	Calibration Value Calibration Value

35.4.20 ADC Plus-Side General Calibration Value Register (ADC_CLPX)

For more information, see CLPD register description.

Address: 0h base + C8h offset = C8h



* Notes:
- CLPX field: Reset values are loaded out of IFR.

ADC_CLPX field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CLPXEN	CLPX compare bit Controls how comparison of CLPX is evaluated to be a pass or fail.
CLPX	Calibration Value Calibration Value

Assignment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

35.4.21 ADC Plus-Side General Calibration Value Register (ADC_CLP9)

For more information, see CLPD register description.

Address: 0h base + CCh offset = CCh

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								CLP9EN	CLP9						
W																
Reset	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*

* Notes:
- CLP9 field: Reset values are loaded out of IFR.

ADC_CLP9 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CLP9EN	CLP9 compare bit Controls how comparison of CLP9 is evaluated to be a pass or fail.
CLP9	Calibration Value Calibration Value

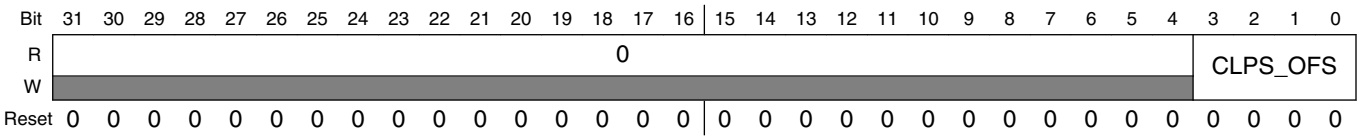
35.4.22 ADC General Calibration Value Register (ADC_CLPS_OFS)

The 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[7:0], CLP1[8:0], CLP2[9:0], CLP3[9:0], CLPS[6:0], CLP9[6:0] and CLPX[6:0]. 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 Asset Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + D0h offset = D0h



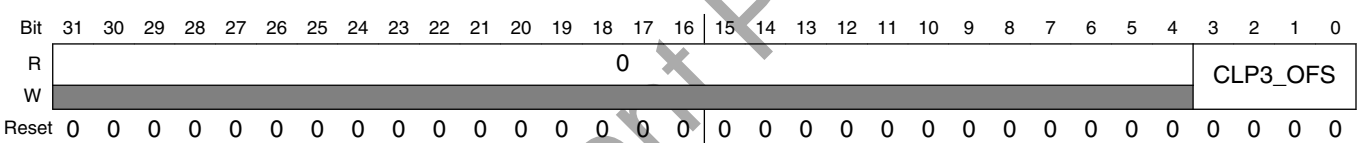
ADC_CLPS_OFS field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLPS_OFS	CLPS Offset Capacitor offset correction value

35.4.23 ADC Plus-Side General Calibration Value Register (ADC_CLP3_OFS)

For more information, see CLP3 register description.

Address: 0h base + D4h offset = D4h



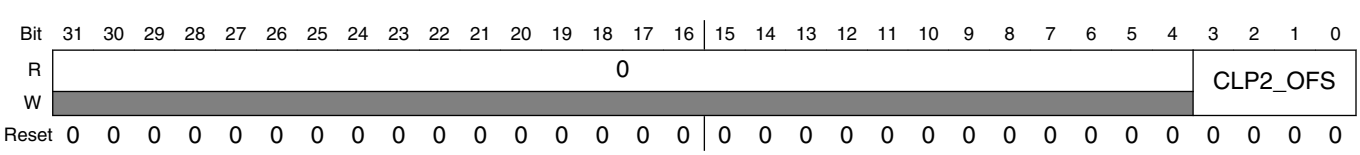
ADC_CLP3_OFS field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP3_OFS	CLP3 Offset Capacitor offset correction value

35.4.24 ADC Plus-Side General Calibration Value Register (ADC_CLP2_OFS)

For more information, see CLP2 register description.

Address: 0h base + D8h offset = D8h



For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ADC_CLP2_OFS field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP2_OFS	CLP2 Offset Capacitor offset correction value

35.4.25 ADC Plus-Side General Calibration Value Register (ADC_CLP1_OFS)

For more information, see CLP1 register description.

Address: 0h base + DCh offset = DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CLP1_OFS																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADC_CLP1_OFS field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP1_OFS	CLP1 Offset Capacitor offset correction value

35.4.26 ADC Plus-Side General Calibration Value Register (ADC_CLP0_OFS)

For more information, see CLPD register description.

Address: 0h base + E0h offset = E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CLP0_OFS																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADC_CLP0_OFS 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...

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

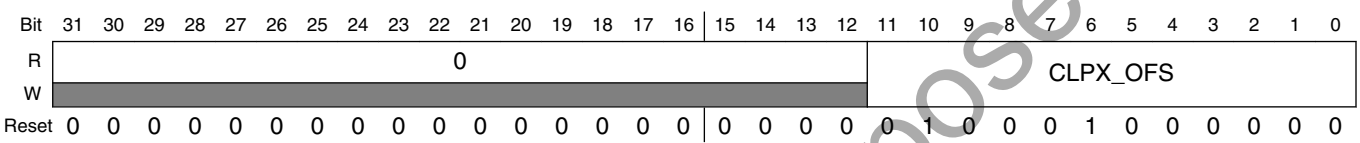
ADC_CLP0_OFS field descriptions (continued)

Field	Description
CLP0_OFS	CLP0 Offset Capacitor offset correction value

35.4.27 ADC Plus-Side General Calibration Value Register (ADC_CLPX_OFS)

For more information, see CLPX_OFS register description.

Address: 0h base + E4h offset = E4h



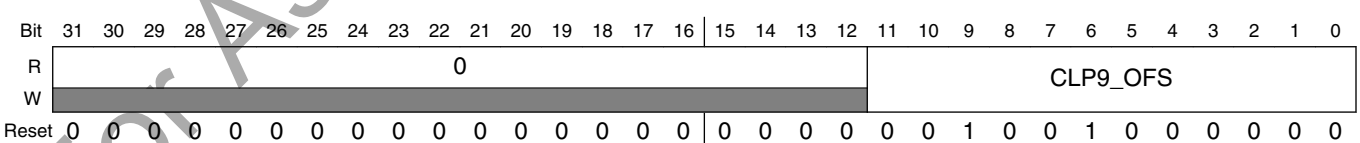
ADC_CLPX_OFS field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLPX_OFS	CLPX Offset Capacitor offset correction value

35.4.28 ADC Plus-Side General Calibration Value Register (ADC_CLP9_OFS)

For more information, see CLP9_OFS register description.

Address: 0h base + E8h offset = E8h



ADC_CLP9_OFS field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP9_OFS	CLP9 Offset Capacitor offset correction value

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

ADC_CLP9_OFS field descriptions (continued)

Field	Description

35.5 Functional description

The ADC module is disabled during reset, 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 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.

35.5.1 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].

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- ALTCLK*x*: As defined for this MCU. See the chip configuration information. Conversions are possible using ALTCLK*x* as the input clock source while the MCU is in Normal Stop mode. ALTCLK1 is the default selection following reset.

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.

35.5.2 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}). These voltage references are selected using SC2[REFSEL]. The alternate V_{ALTH} voltage reference may select additional external pin or internal source depending on MCU configuration. See the chip configuration information on the voltage references specific to this MCU.

35.5.3 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

35.5.4 Conversion control

Conversions can be performed as determined by CFG1[MODE] 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

35.5.4.1 Initiating conversions

A conversion is initiated:

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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.

35.5.4.2 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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

35.5.4.3 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.

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, RA and Rn return to their reset states.

35.5.4.4 Power control

The ADC module remains in its idle state until a conversion is initiated.

35.5.4.5 Sample time and total conversion time

The total conversion time depends upon:

- The sample time as determined by CFG2[SMPLTS]
- The MCU bus frequency
- The conversion mode, as determined by CFG1[MODE]
- The frequency of the conversion clock, that is, f_{ADCK}.

After the module becomes active, sampling of the input begins.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

1. CFG2[SMPLTS] selects 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.

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].

35.5.4.6 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.

35.5.5 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 35-4. Compare modes

SC2[ACFGT]	SC2[ACREN]	CV1 relative to CV2	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.

For Assessment P

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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

35.5.6 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 calibrated gain value, and each of the calibration values. The offset calibration value is automatically stored in the ADC offset correction register (OFS), the calibration gain value is stored in the ADC calibration gain register (G) and the calibration values are automatically stored in each calibration registers, CLPx. The user must configure the ADC correctly prior to calibration, and must generate the user offset and store them in the ADC user offset (USR_OFS) 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 60 MHz
- $V_{REFH}=V_{DDA}$
- Calibrate at nominal voltage and temperature

The input channel, conversion mode continuous function, compare function, resolution mode, and 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 . 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 . 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.

If calibration values or user gain register are over-written after calibration, prior to next conversion, the user must generate a new gain calibration values using the following procedure:

1. Initialize or clear a 16-bit variable in RAM.
2. Add the new calibration values UG, CLP0, CLP1, CLP2, CLP3, and CLPS to the variable.

For Assessment Purposes Only

3. If OR of MSB[15:11] of the variable is set overwrite and set variable to 16'hffff.
4. Store the value in the gain calibration register G.

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 some time, depending on the clock source chosen and the average time configuration. To reduce this latency, the calibration values, which are offset, gain, and 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 7 register store operations on all subsequent power, reset, or Low-Power Stop mode recoveries.

35.5.7 User-defined offset function

OFS contain the 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.

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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The offset calibration function may be employed by the user to remove application offsets or DC bias values. USR_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.

35.5.8 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 Alternate Clock sources 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.

35.5.9 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.

For Assessment Purposes Only

35.5.9.1 Normal Stop mode with Alternate clock sources enabled

If Alternate clock source selected for the conversion clock is enabled, 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 36
Comparator (CMP)

36.1 Chip-specific Comparator (CMP) information

36.1.1 Instantiation information

There is 1 analog comparator (CMP0) on this device. DAC output is not supported for applications external to DAC on this device.

- CMP has its own independent 8-bit DAC.
- CMP supports up to 8 analog inputs from external pins.
- CMP is able to convert an internal reference from the bandgap (1V reference voltage).
- CMP0 supports the round-robin sampling scheme. In summary, this allow the CMP to operate independently in STOP and VLPS mode, whilst being triggered periodically to sample up to 8 inputs. Only if an input changes state is a full wakeup generated.

36.1.2 CMP input connections

The following table shows the input connections to the CMP.

Table 36-1. CMP input connections

CMP Inputs	CMP0
IN0	CMP0_IN0
IN1	CMP0_IN1
IN2	CMP0_IN2
IN3	CMP0_IN3
IN4	CMP0_IN4

Table continues on the next page...

Assessment Purposes Only

Table 36-1. CMP input connections (continued)

CMP Inputs	CMP0
IN5	CMP0_IN5
IN6	CMP0_IN6
IN7	CMP0_IN7

36.1.3 CMP external references

The CMP could get external reference through the tightly integrated 8-bit DAC sub-block. The 8-bit DAC sub-block supports selection of two references. For this device, the references are connected as follows:

- VDDA
- PMC bandgap buffer out (1V reference voltage)

36.1.4 External window/sample input

PDB and LPIT could be used to generate pulse output which can be used as sampling windows of CMP block via TRGMUX.

36.1.5 CMP trigger mode

The CMP and 8-bit DAC sub-block supports trigger mode operation when the chip is in STOP or VLPS mode. When trigger mode is enabled, the trigger source will provide a low power clock and the triggers to the CMP. The trigger event will initiate a compare sequence that must first enable the CMP and DAC prior to performing a CMP operation and capturing the output. In this device, control for this two staged sequencing is provided from the LPTMR. The LPTMR triggering output is always enabled when the LPTMR is enabled. The first signal is supplied to enable the CMP and DAC and is asserted at the same time as the TCF flag is set. The delay to the second signal that triggers the CMP to capture the result of the compare operation is dependent on the LPTMR configuration. In Time Counter mode with prescaler enabled, the delay is 1/2 Prescaler output period. In Time Counter mode with prescaler bypassed, the delay is 1/2 Prescaler source 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

36.2 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 8-bit digital-to-analog converter (DAC). The mux circuit is designed to operate across the full range of the supply voltage.

The DAC is a 256-tap resistor ladder network that provides a selectable voltage reference for applications requiring a voltage reference. The 256-tap resistor ladder network divides the supply reference V_{in} into 256 voltage levels. A 8-bit digital signal input selects the output voltage level, which varies from V_{in} to $V_{in}/256$. V_{in} can be selected from two voltage sources, V_{in1} and V_{in2}. The DAC from a comparator is available as an on-chip internal signal only and is not available externally to a pin.

36.3 Features

The following subsections list the features of the CMP, the DAC, and the ANMUX.

36.3.1 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
  - Windowed, which is ideal for certain PWM zero-crossing-detection applications
  - Digitally filtered:

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

    - Filter can be bypassed
    - Can be clocked via external SAMPLE signal or 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 power modes available on this MCU
- The window and filter functions are not available in STOP modes
- The comparator can be triggered by other peripherals to work for only a small fraction of the time

36.3.2 8-bit DAC key features

The DAC has the following features:

- 8-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

36.3.3 ANMUX key features

The ANMUX has the following features:

- Two 8-to-1 channel MUXes
- Operational over the entire supply range

For Assessment Purposes Only

36.4 CMP, DAC, and ANMUX diagram

The following figure shows the block diagram for the High-Speed Comparator, DAC, and ANMUX modules.

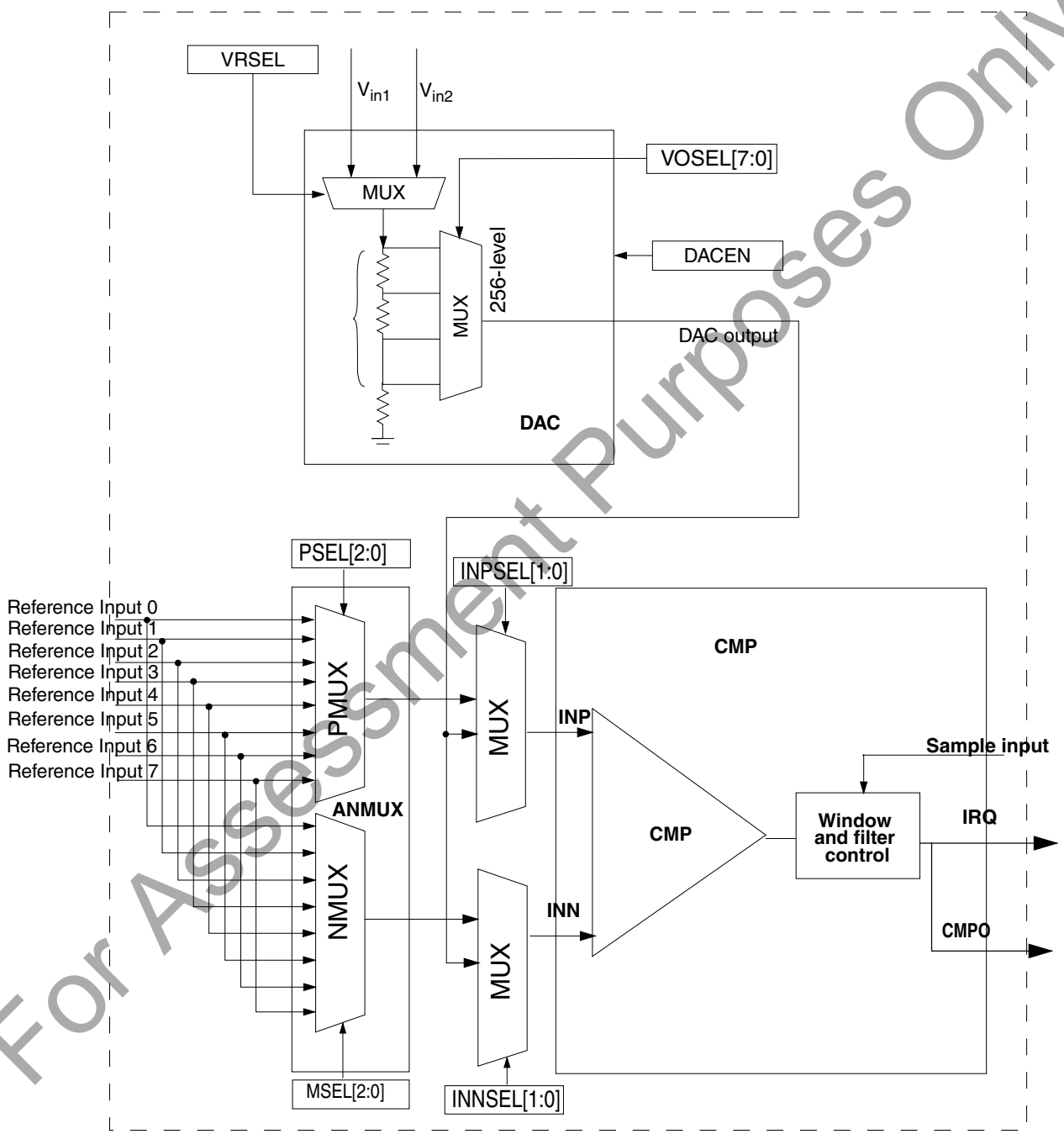


Figure 36-1. CMP high level diagram

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

36.5 CMP block diagram

The following figure shows the block diagram for the CMP module.

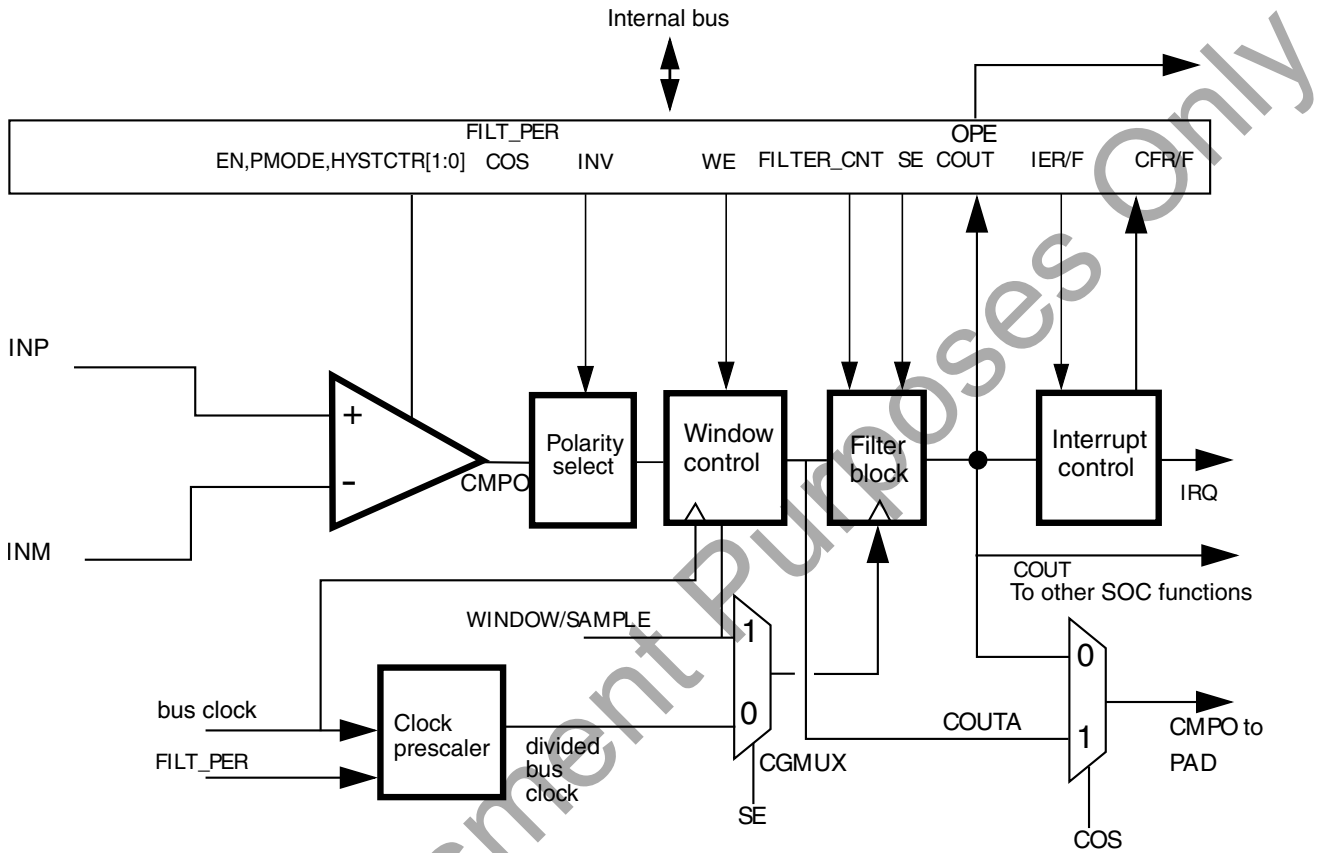


Figure 36-2. Comparator module block diagram

In the CMP block diagram:

- The Window Control block is bypassed when C0[WE] = 0.
- If C0[WE] = 1, the comparator output is sampled on every bus clock when WINDOW=1 to generate COUTA. Sampling does NOT occur when WINDOW = 0.
- The Filter block is bypassed when not in use.

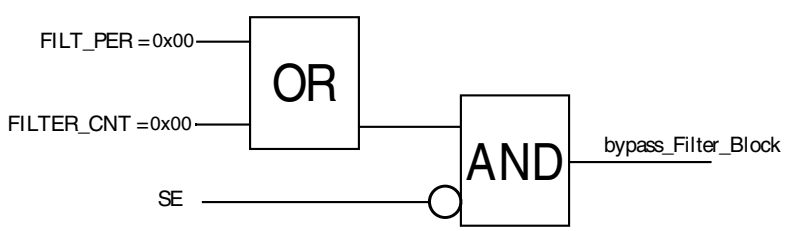


Figure 36-3. Filter block bypass logic

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- The Filter block acts as a simple sampler if the filter is bypassed and C0[FILTER_CNT] is set to 0x01.
- The Filter block filters based on multiple samples when the filter is bypassed and C0[FILTER_CNT] is set greater than 0x01.
  - If C0[SE] = 1, the external SAMPLE input is used as the sampling clock.
  - IF C0[SE] = 0, the divided bus clock is used as the 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 crosses clock domain boundaries, must be resynchronized to the bus clock.
- C0[WE] and C0[SE] are mutually exclusive.
- If enabled, the filter clock and the sample period must be at least 4 tims slower than the system clock to the comparator.

36.6 CMP pin descriptions

This section provides the comparator pin descriptions. The external inputs IN[7:0] are muxed by CMP_C1[PSEL] and CMP_C1[MSEL] beforehand and multiplexed output will then go to the second stage of multiplex with the input of 8-bit DAC and other two internal reserved test signals, determined by CMP_C1[INPSEL] and CMP_C1[INNSEL] the output of the second multiplex will finall go to the positive and negative ports of the comparator respectively.

Table 36-2. CMP signal descriptions

Signal	Description	I/O
IN[7:0]	Analog voltage inputs	I

36.6.1 External pins

The CMP has two analog inputs: INP and INM. Each of these pins can accept an input voltage that varies across the full operating range of the MCU. If the module is not enabled, each pin can be used as a digital input or output. Consult the specific MCU documentation to determine what functions are shared with these analog inputs.

The user can select either filtered or unfiltered comparator outputs for use on an external I/O pad.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

36.7 CMP functional modes

There are three main sub-blocks to the CMP module:

- The comparator itself
- The window function
- The filter function

The filter, C0[FILTER_CNT], can be clocked from an internal or external clock source. 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 external sample input is enabled using C0[SE]. When set, the output of the comparator is sampled only on rising edges of the sample input.

The "windowing mode" is enabled by setting C0[WE]. When set, the comparator output is sampled only when WINDOW=1. This feature can be used to ignore the comparator output during time periods in which the input voltages are not valid. This is especially useful when implementing zero-crossing-detection for certain PWM applications.

The comparator filter and sampling features can be combined as shown in the following table. Individual modes are discussed below.

ment Purposes Only

Table 36-3. Comparator sample/filter controls

Mode #	C0[EN]	C0[WE]	C0[SE]	C0[FILTER_CNT]	C0[FPR]	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	
3A	1	0	1	0x01	X	**Sampled, Non-Filtered mode** See the Sampled, Non-Filtered mode (#s 3A & 3B).
3B	1	0	0	0x01	> 0x00	
4A	1	0	1	> 0x01	X	**Sampled, Filtered mode** See the Sampled, Filtered mode (#s 4A & 4B).
4B	1	0	0	> 0x01	> 0x04	
5A	1	1	0	0x00	X	**Windowed mode**
5B	1	1	0	X	0x00	

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 36-3. Comparator sample/filter controls (continued)

Mode #	C0[EN]	C0[WE]	C0[SE]	C0[FILTER_CNT]	C0[FPR]	Operation
						Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA. See the Windowed mode (#s 5A & 5B).
6	1	1	0	0x01	0x01–0xFF	**Windowed/Resampled mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA, which is then resampled on an interval determined by C0[FPR] to generate COUT. See the Windowed/Resampled mode (# 6).
7	1	1	0	> 0x01	0x01–0xFF	**Windowed/Filtered mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA, which is then resampled and filtered to generate COUT. See the Windowed/Filtered mode (#7).
All other combinations of C0[EN], C0[WE], C0[SE], C0[FILTER_CNT], and C0[FPR] 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 C0[SE]=0, C0[FPR] =0 and C0[FILTER_CNT]=0x00. This resets the filter to a known state.

For Assessment

36.7.1 Disabled mode (# 1)

In Disabled mode, the analog comparator is non-functional and consumes no power. CMPO is 0 in this mode.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

36.7.2 Continuous mode (#s 2A & 2B)

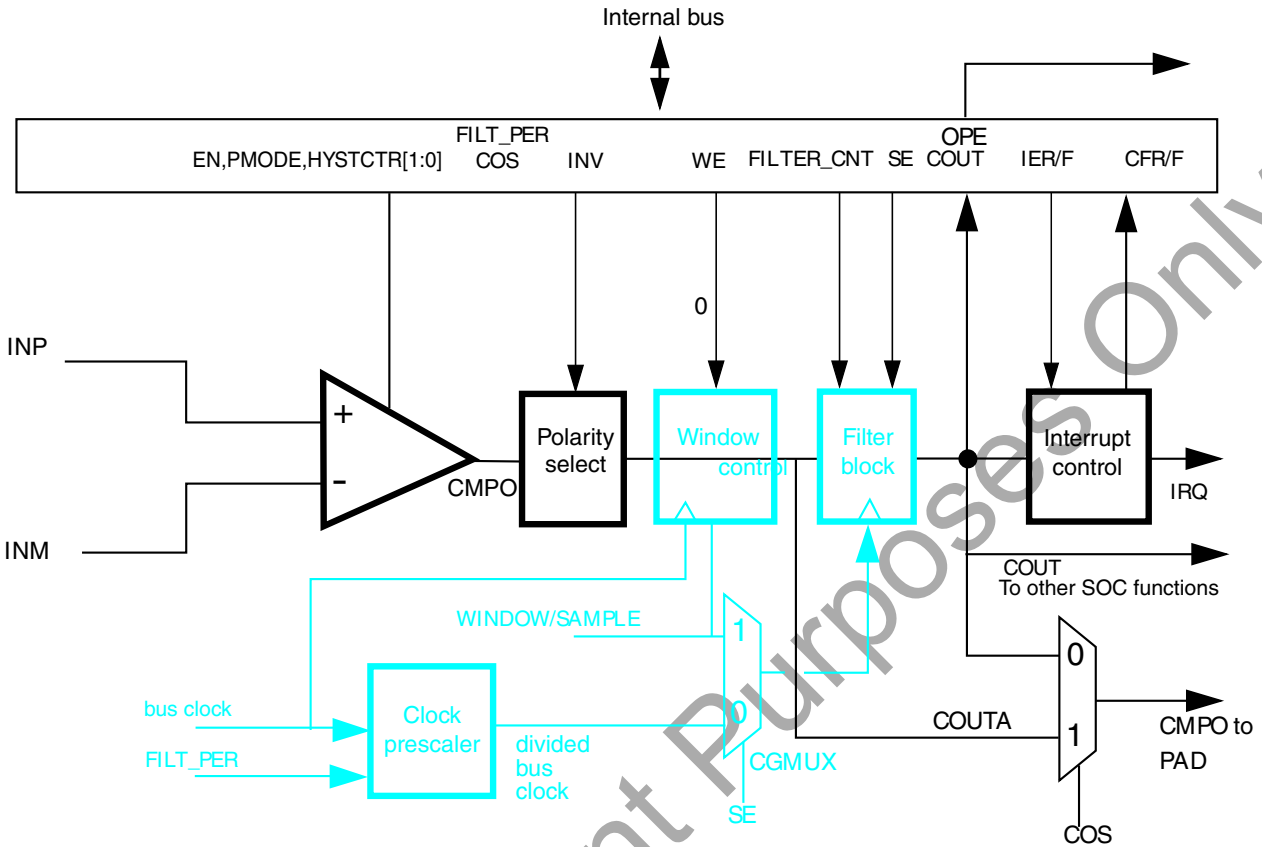


Figure 36-4. Comparator operation in Continuous mode

NOTE

See the chip configuration section for the source of sample/ window input.

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. C0[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 that result in disabling the filter block, see Figure 36-3.

36.7.3 Sampled, Non-Filtered mode (#s 3A & 3B)

For Assessment Purposes Only

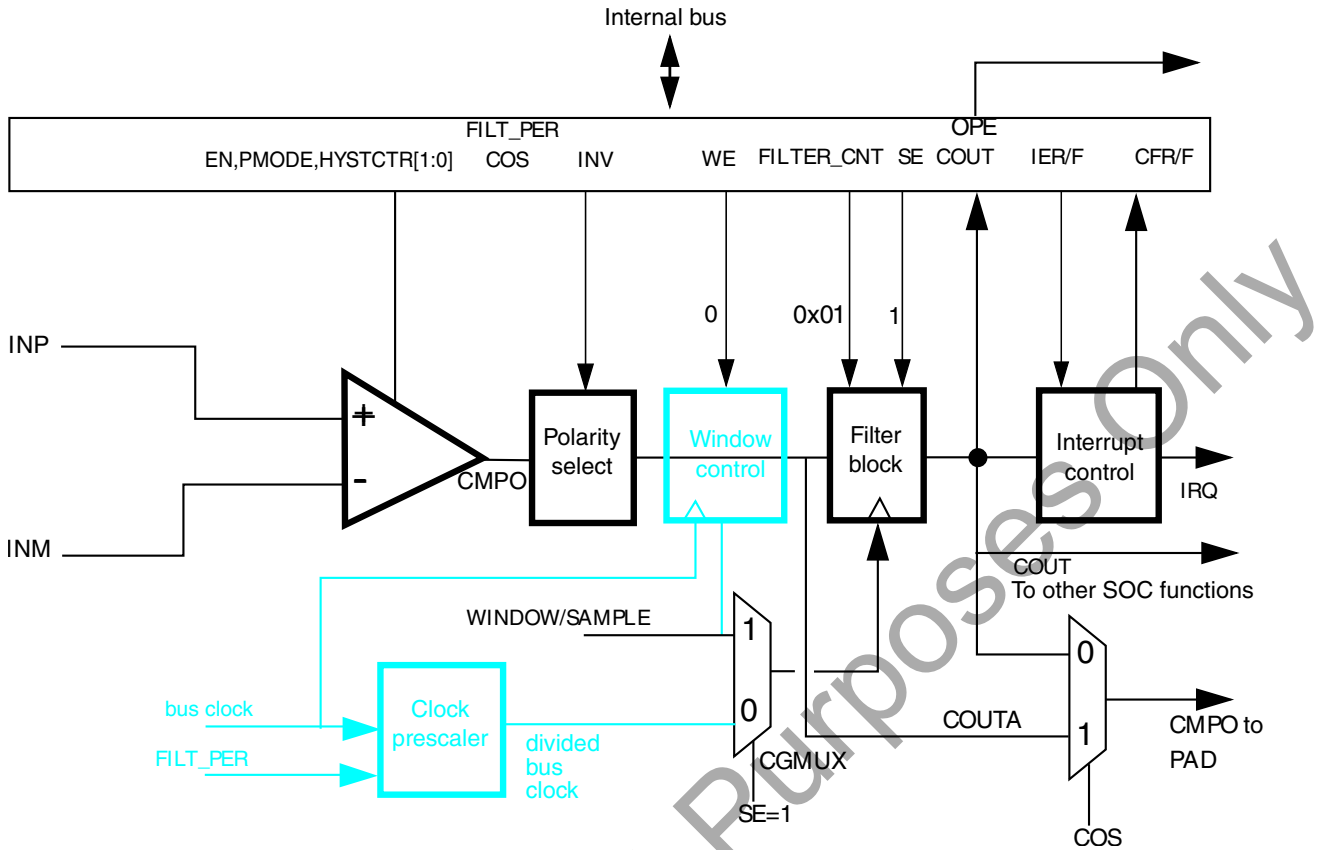


Figure 36-5. Sampled, Non-Filtered (# 3A): sampling point externally driven

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 only difference in operation between Sampled, Non-Filtered (# 3A) and Sampled, Non-Filtered (# 3B) is in how the clock to the filter block is derived. In #3A, the clock to filter block is externally derived while in #3B, the clock to filter block is internally derived.

The comparator filter has no other function than sample/hold of the comparator output in this mode (# 3B).

For Assessment Purposes Only

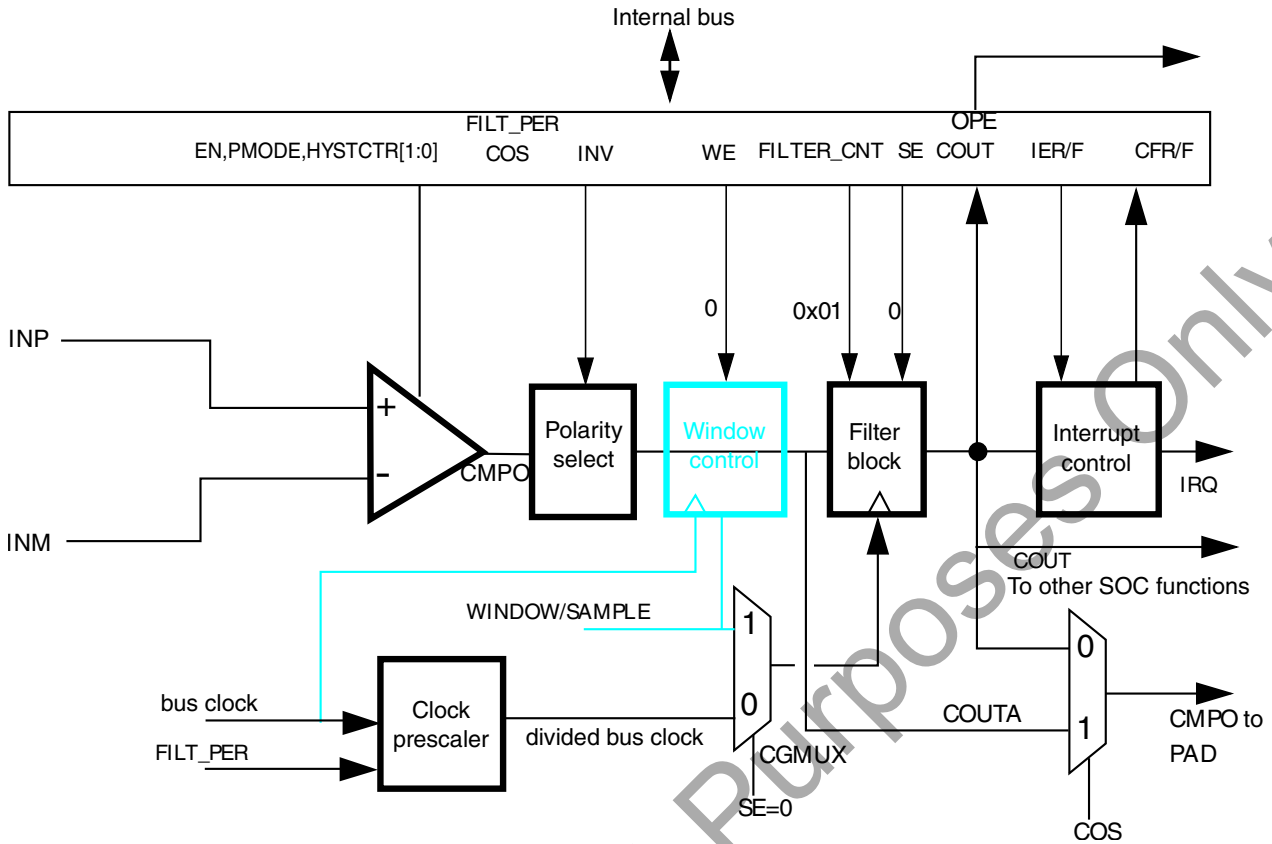


Figure 36-6. Sampled, Non-Filtered (# 3B): sampling interval internally derived

36.7.4 Sampled, Filtered mode (#s 4A & 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.

The only difference in operation between Sampled, Non-Filtered (# 3A) and Sampled, Filtered (# 4A) is that, now, C0[FILTER_CNT]>1, which activates filter operation.

For Assessment Purposes Only

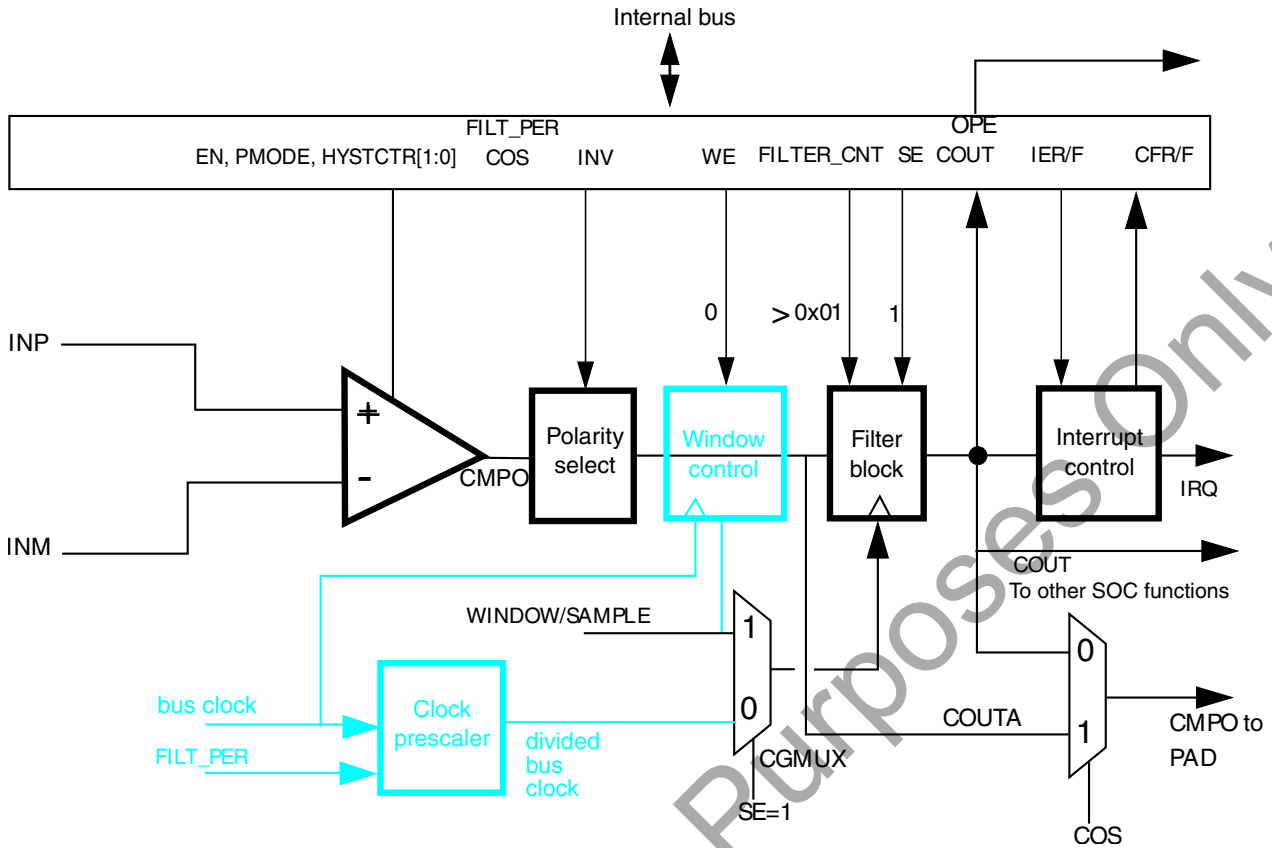


Figure 36-7. Sampled, Filtered (# 4A): sampling point externally driven

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

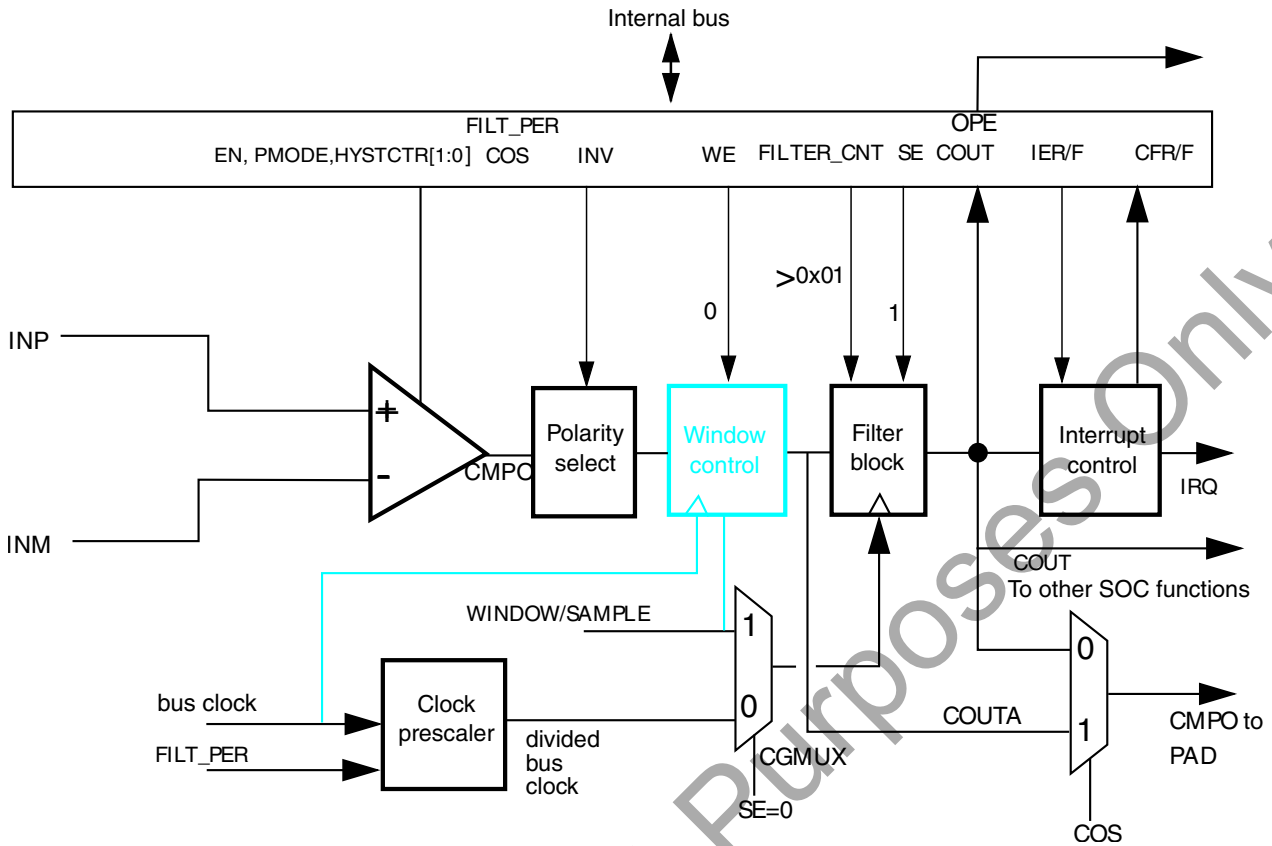


Figure 36-8. Sampled, Filtered (# 4B): sampling point internally derived

The only difference in operation between Sampled, Non-Filtered (# 3B) and Sampled, Filtered (# 4B) is that now, C0[FILTER_CNT]>1, which activates filter operation.

36.7.5 Windowed mode (#s 5A & 5B)

The following figure illustrates comparator operation in the Windowed mode, ignoring latency of the analog comparator, polarity select, and window control block. It also assumes that the polarity select is set to non-inverting state.

NOTE

The analog comparator output is passed to COUTA only when the WINDOW signal is high.

In actual operation, COUTA may lag the analog inputs by up to one bus clock cycle plus the combinational path delay through the comparator and polarity select logic.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

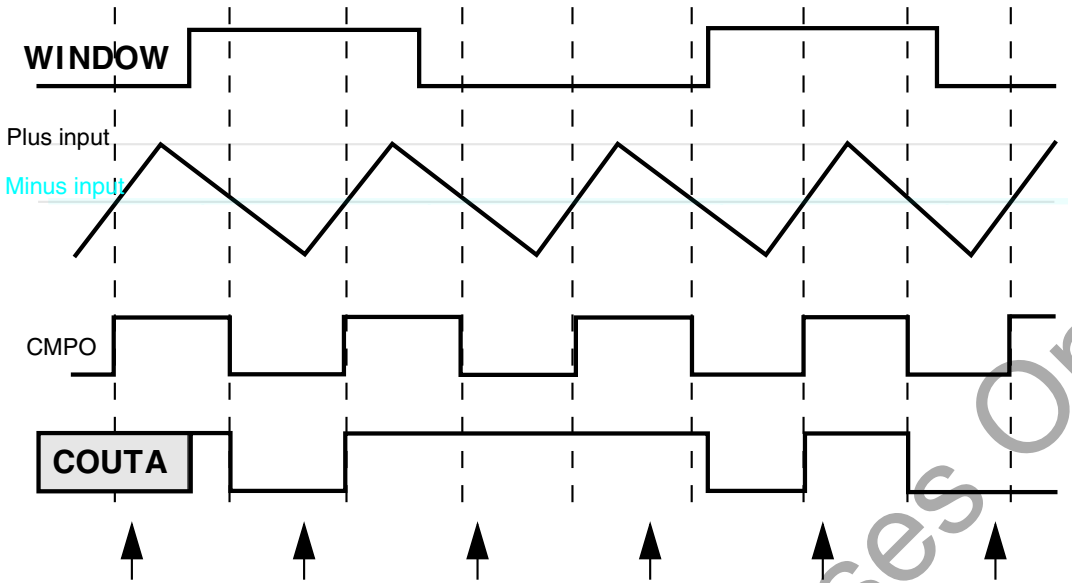


Figure 36-9. Windowed mode timing diagram

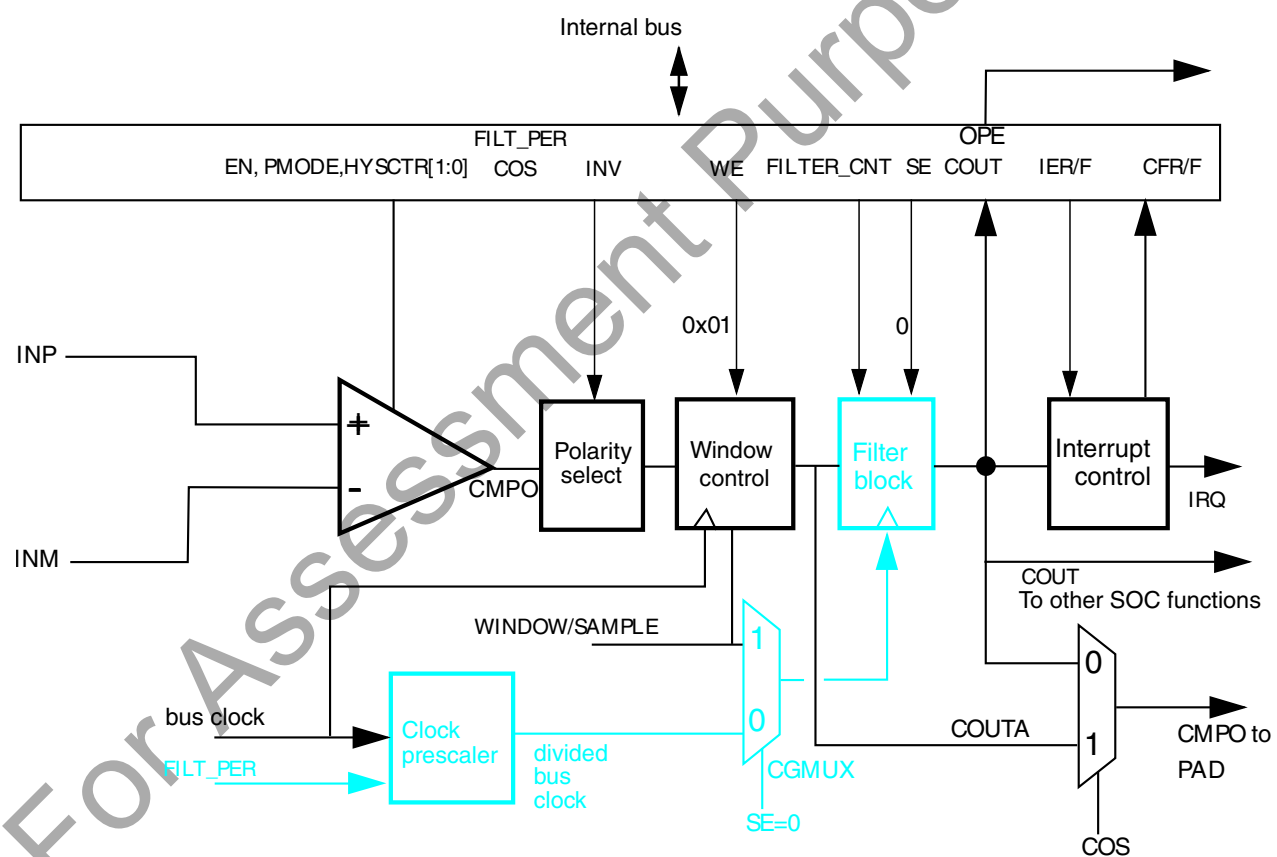


Figure 36-10. Windowed mode

For control configurations which result in disabling the filter block, see Figure 36-3.

When any windowed mode is active, COUTA is clocked by the bus clock whenever WINDOW = 1. The last latched value is held when WINDOW = 0.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

36.7.6 Windowed/Resampled mode (# 6)

The following figure uses the same input stimulus shown in Figure 36-9, and adds resampling of COUTA to generate COUT. Samples are taken at the time points indicated by the arrows in the figure. Again, prop delays and latency are ignored for the sake of clarity.

This example was generated solely to demonstrate operation of the comparator in windowed/resampled mode, and does not reflect any specific application. Depending upon the sampling rate and window placement, COUT may not see zero-crossing events detected by the analog comparator. Sampling period and/or window placement must be carefully considered for a given application.

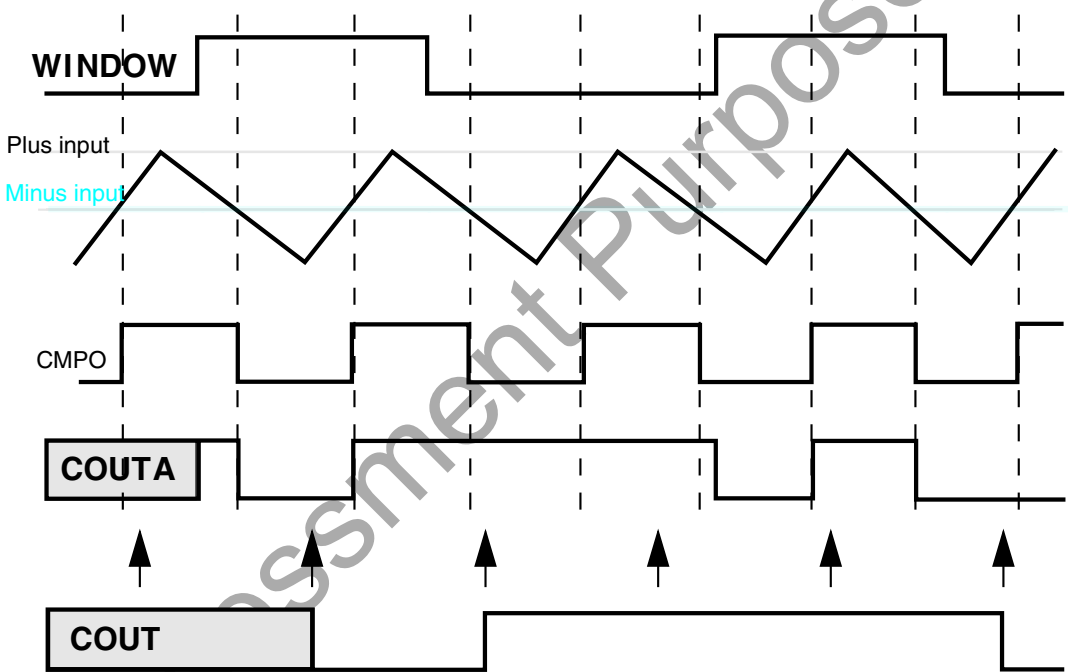


Figure 36-11. Windowed/resampled mode operation

This mode of operation results in an unfiltered string of comparator samples where the interval between the samples is determined by FPR[FILT_PER] and the bus clock rate. Configuration for this mode is virtually identical to that for the Windowed/Filtered Mode shown in the next section. The only difference is that the value of C0[FILTER_CNT] must be 1.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

36.7.7 Windowed/Filtered mode (#7)

This is the most complex mode of operation for the comparator block, as it uses both windowing and filtering features. It also has the highest latency of any of the modes. This can be approximated: up to 1 bus clock synchronization in the window function + ((C0[FILTER_CNT] * C0[FPR]) + 1) * bus clock for the filter function.

When any windowed mode is active, COUTA is clocked by the bus clock whenever WINDOW = 1. The last latched value is held when WINDOW = 0.

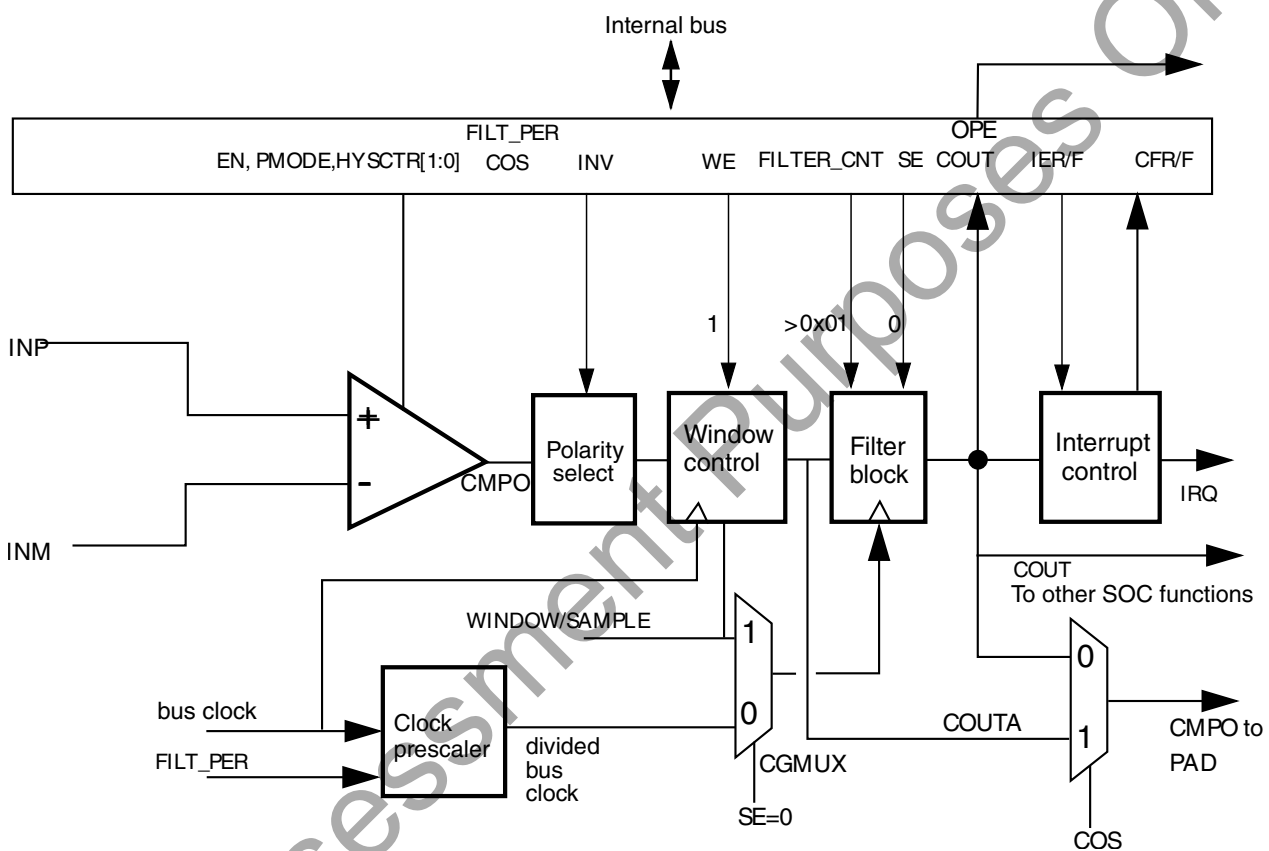


Figure 36-12. Windowed/Filtered mode

36.8 Memory map/register definitions

CMP memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	CMP Control Register 0 (CMP_C0)	32	R/W	0000_0000h	36.8.1/744
4	CMP Control Register 1 (CMP_C1)	32	R/W	0000_0000h	36.8.2/747
8	CMP Control Register 2 (CMP_C2)	32	R/W	0000_0000h	36.8.3/750

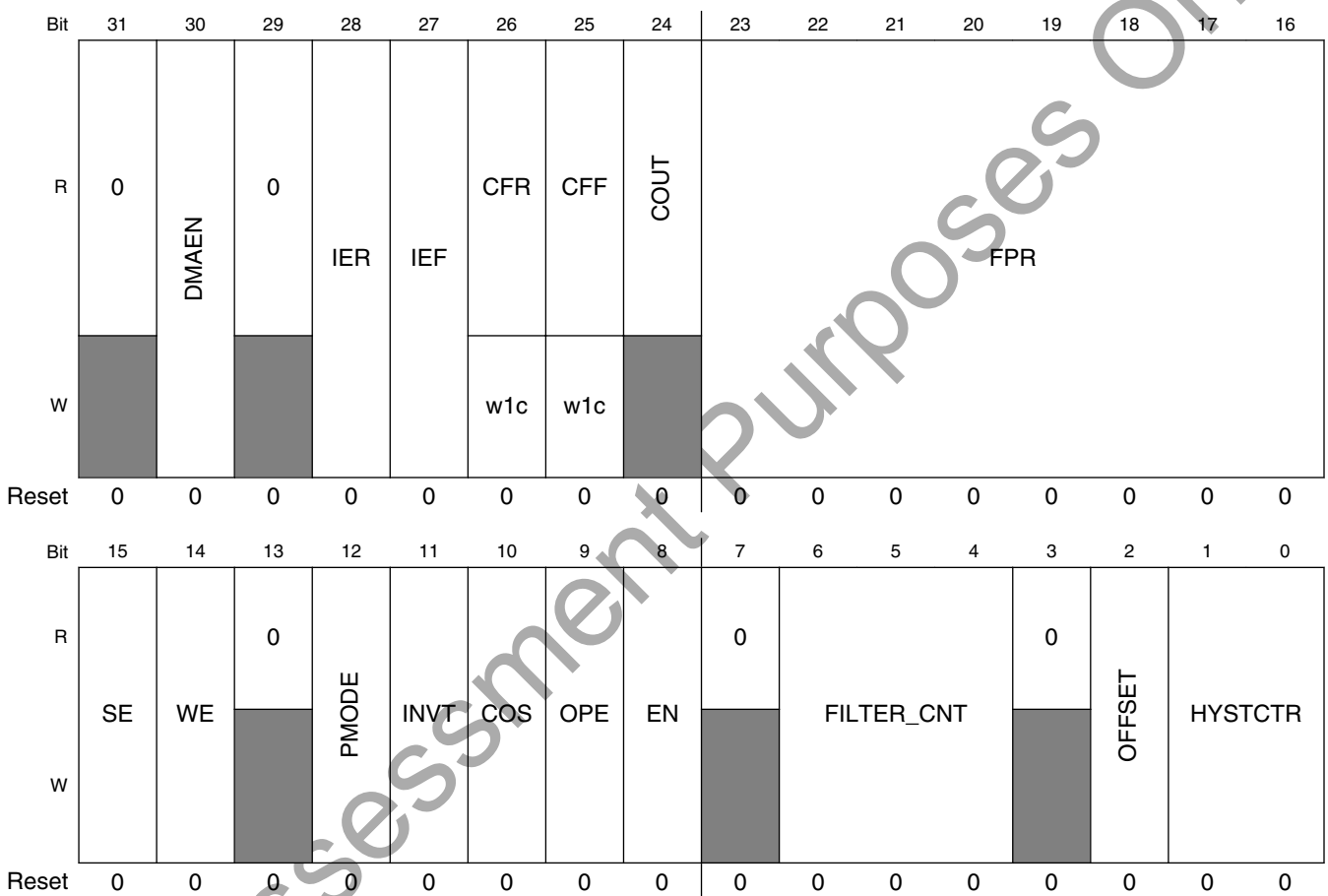
For Assessment Purposes Only

36.8.1 CMP Control Register 0 (CMP_C0)

Access:

- Supervisor read/write
- User read/write

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



Assessment Purposes Only

CMP_C0 field descriptions

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

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CMP_C0 field descriptions (continued)

Field	Description
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 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.
27 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.
26 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 A rising edge has not been detected on COUT. 1 A rising edge on COUT has occurred.
25 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 senstive . 0 A falling edge has not been detected on COUT. 1 A falling edge on COUT has occurred.
24 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 C0[INVT] when the Analog Comparator module is disabled, that is, when C0[EN] = 0. Writes to this field are ignored.
23–16 FPR	Filter Sample Period Specifies the sampling period, in bus clock cycles, of the comparator output filter, when C0[SE] = 0. Setting FPR to 0x0 disables the filter. Filter programming and latency details are provided in the CMP functional description. This field has no effect when C0[SE]= 1. In that case, the external SAMPLE signal is used to determine the sampling period.
15 SE	Sample Enable At any given time, either SE or WE can be set. If a write to this register attempts to set both, then SE is set and WE is cleared. However, avoid writing ones to both bit locations because this "11" case is reserved. 0 Sampling mode is not selected. 1 Sampling mode is selected.
14 WE	Windowing Enable At any given time, either SE or WE can be set. If a write to this register attempts to set both, then SE is set and WE is cleared. However, avoid writing ones to both bit locations because this "11" case is reserved.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CMP_C0 field descriptions (continued)

Field	Description
	0 Windowing mode is not selected. 1 Windowing mode is selected.
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 PMODE	Power Mode Select 0 Low Speed (LS) comparison mode is selected. 1 High Speed (HS) comparison mode is selected, in VLPx mode, or Stop mode switched to Low Speed (LS) mode.
11 INVT	Comparator invert This bit allows selecting the polarity of the analog comparator function. It is also driven to the COUT output (on both the device pin and as C0[COUT]) when C0[OPE]=0. 0 Does not invert the comparator output. 1 Inverts the comparator output.
10 COS	Comparator Output Select 0 Set CMPO to equal COUT (filtered comparator output). 1 Set CMPO to equal COUTA (unfiltered comparator output).
9 OPE	Comparator Output Pin Enable The OPE bit enables the path from the comparator output to a selected pin. 0 When OPE is 0, the comparator output (after window/filter settings dependent on software configuration) is not available to a packaged pin. 1 When OPE is 1, and if the software has configured the comparator to own a packaged pin, the comparator is available in a packaged pin.
8 EN	Comparator Module Enable The EN bit enables the Analog Comparator Module. When the module is not enabled, the analog part remains in the off state, and consumes no power. When the same input is selected from analog mux to the positive and negative port, the comparator is disabled automatically. 0 Analog Comparator is disabled. 1 Analog Comparator is enabled.
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 This field specifies 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, please see the Functional Description. 000 Filter is disabled. If SE = 1, then COUT is a logic zero (this is not a legal state, and is not recommended). If SE = 0, COUT = COUTA. 001 1 consecutive sample must agree (comparator output is simply sampled). 010 2 consecutive samples must agree. 011 3 consecutive samples must agree. 100 4 consecutive samples must agree.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CMP_C0 field descriptions (continued)

Field	Description
	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 OFFSET	Comparator hard block offset control. See chip data sheet to get the actual offset value with each level **NOTE:** • If OFFSET = 1, then there will be no hysteresis in the case of INP crossing INN in the positive direction (or INN crossing INP in the negative direction). A Half Hysteresis value still exists for INP crossing INN in the falling direction. • If OFFSET = 0, then the hysteresis selected by HYSTCTR is valid for both directions. 0 The comparator hard block output has level 0 offset internally. 1 The comparator hard block output has level 1 offset internally.
HYSTCTR	Comparator hard block hysteresis control. See chip data sheet to get the actual hystersis value with each level 00 The hard block output has level 0 hysteresis internally. 01 The hard block output has level 1 hysteresis internally. 10 The hard block output has level 2 hysteresis internally. 11 The hard block output has level 3 hysteresis internally.

36.8.2 CMP Control Register 1 (CMP_C1)

Access:

- Supervisor read/write
- User read/write

Address: 0h base + 4h offset = 4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0	INPSEL		0	INNSEL		CHN7	CHN6	CHN5	CHN4	CHN3	CHN2	CHN1	CHN0
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	DACEN	VRSEL	PSEL			MSEL			VOSEL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CMP_C1 field descriptions

Field	Description
31–30 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.
28–27 INPSEL	Selection of the input to the positive port of the comparator Determines which input is selected for the plus input of the comparator. NOTE: These selections is used to select the final positive input to the comparator. Note: For the round robin mode of operation, the MSEL and PSEL bitfields in CMP*x*_C1 register must have different values. 00 IN0, from the 8-bit DAC output 01 IN1, from the analog 8-1 mux 10 Reserved 11 Reserved
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 INNSEL	Selection of the input to the negative port of the comparator Determines which input is selected for the plus input of the comparator. NOTE: These selections is used to select the final negative input to the comparator. Note: For the round robin mode of operation, the MSEL and PSEL bitfields in CMP*x*_C1 register must have different values. 00 IN0, from the 8-bit DAC output 01 IN1, from the analog 8-1 mux 10 Reserved 11 Reserved
23 CHN7	Channel 7 input enable Channel 7 of the input enable for the round-robin checker. If CHN7 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
22 CHN6	Channel 6 input enable Channel 6 of the input enable for the round-robin checker. If CHN6 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
21 CHN5	Channel 5 input enable Channel 5 of the input enable for the round-robin checker. If CHN5 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
20 CHN4	Channel 4 input enable Channel 4 of the input enable for the round-robin checker. If CHN4 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
19 CHN3	Channel 3 input enable

Table continues on the next page...

Review Draft
Freescale Confidential Proprietary - Non-Disclosure Agreement required

CMP_C1 field descriptions (continued)

Field	Description
	Channel 3 of the input enable for the round-robin checker. If CHN3 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
18 CHN2	Channel 2 input enable Channel 2 of the input enable for the round-robin checker. If CHN2 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
17 CHN1	Channel 1 input enable Channel 1 of the input enable for the round-robin checker. If CHN1 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
16 CHN0	Channel 0 input enable Channel 0 of the input enable for the round-robin checker. If CHN0 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
15 DACEN	DAC Enable This bit is used to enable the DAC. When the DAC is disabled, it is powered down to conserve power. 0 DAC is disabled. 1 DAC is enabled.
14 VRSEL	Supply Voltage Reference Source Select 0 Vin1 is selected as resistor ladder network supply reference Vin. 1 Vin2 is selected as resistor ladder network supply reference Vin.
13–11 PSEL	Plus Input MUX Control Determines which input is selected for the plus mux. NOTE: These bits are used to selecte the external 8 inputs for the plus mux, the actual input to the positive port of the comparator is selected between this mux out and other inputs finally, see the definition in INPSEL. Note: For the round robin mode of operation, the MSEL and PSEL bitfields in CMPx_C1 register must have different values. 000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7
10–8 MSEL	Minus Input MUX Control Determines which input is selected for the minus mux. NOTE: These bits are used to select the external 8 inputs for the minus mux, the actual input to the negative port of the comparator is selected between this mux out and other inputs finally, see the definition in INNSEL.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CMP_C1 field descriptions (continued)

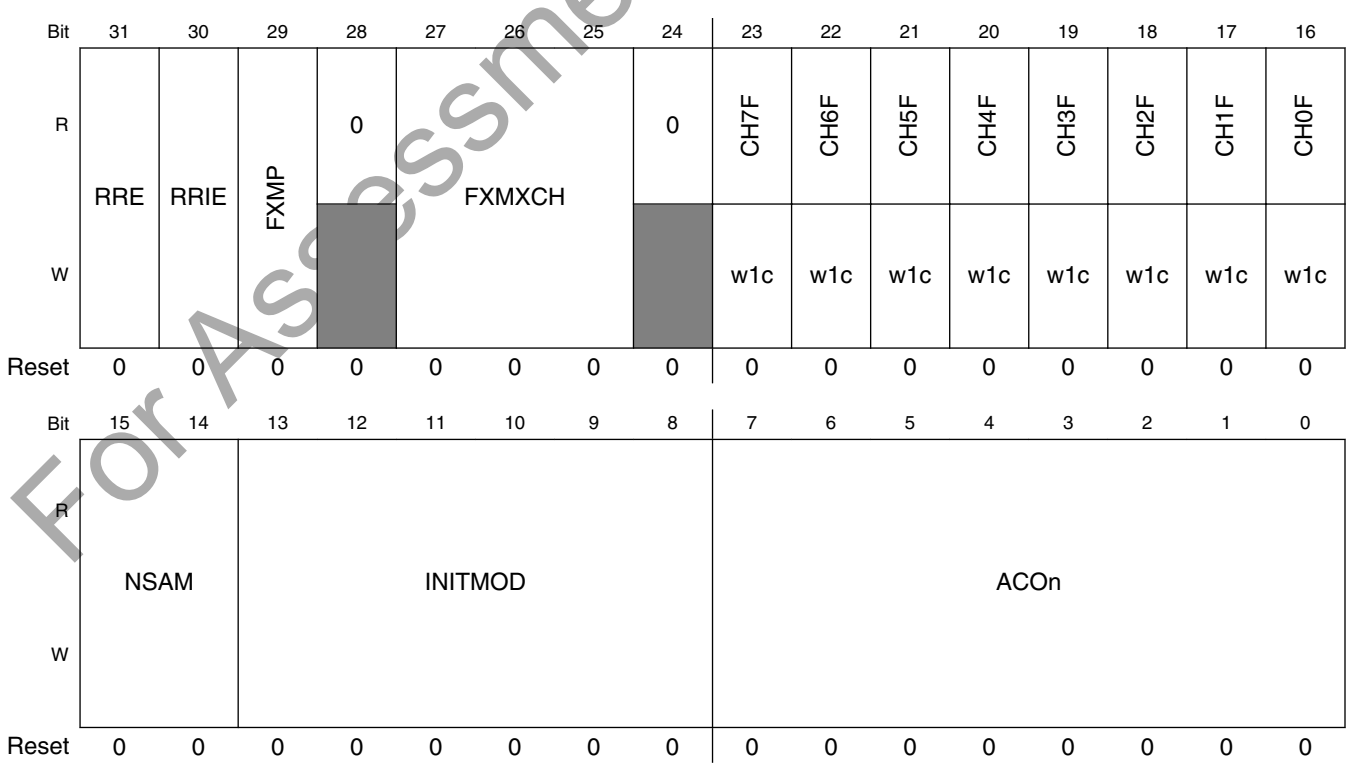
Field	Description
	Note: For the round robin mode of operation, the MSEL and PSEL bitfields in CMP*x*_C1 register must have different values. 000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7
VOSEL	DAC Output Voltage Select This bit selects an output voltage from one of 256 distinct levels. DACO = (Vin/256) * (VOSEL[7:0] + 1), so the DACO range is from Vin/256 to Vin.

36.8.3 CMP Control Register 2 (CMP_C2)

Access:

- Supervisor read/write
- User read/write

Address: 0h base + 8h offset = 8h



For Assessment Purpos

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CMP_C2 field descriptions

Field	Description
31 RRE	Round-Robin Enable This bit enables the round-robin operation. 0 Round-robin operation is disabled. 1 Round-robin operation is enabled.
30 RRIE	Round-Robin interrupt enable This bit enables the interrupt/wake-up when the comparison result changes for a given channel. 0 The round-robin interrupt is disabled. 1 The round-robin interrupt is enabled when a comparison result changes from the last sample.
29 FXMP	Fixed MUX Port This bit is used to fix the analog mux port for the round-robin mode. 0 The Plus port is fixed. Only the inputs to the Minus port are swept in each round. 1 The Minus port is fixed. Only the inputs to the Plus port are swept in each round.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–25 FXMXCH	Fixed channel selection This field indicates which channel in the mux port is fixed in a given round-robin mode. If FXDACI is set, FXMXCH has no effect. 000 Channel 0 is selected as the fixed reference input for the fixed mux port. 001 Channel 1 is selected as the fixed reference input for the fixed mux port. 010 Channel 2 is selected as the fixed reference input for the fixed mux port. 011 Channel 3 is selected as the fixed reference input for the fixed mux port. 100 Channel 4 is selected as the fixed reference input for the fixed mux port. 101 Channel 5 is selected as the fixed reference input for the fixed mux port. 110 Channel 6 is selected as the fixed reference input for the fixed mux port. 111 Channel 7 is selected as the fixed reference input for the fixed mux port.
24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 CH7F	Channel 7 input changed flag. This bit is set If the channel 7 input changed from the last comparison with the fixed mux port.
22 CH6F	Channel 6 input changed flag. This bit is set If the channel 6 input changed from the last comparison with the fixed mux port.
21 CH5F	Channel 5 input changed flag. This bit is set If the channel 5 input changed from the last comparison with the fixed mux port.
20 CH4F	Channel 4 input changed flag. This bit is set If the channel 4 input changed from the last comparison with the fixed mux port.
19 CH3F	Channel 3 input changed flag. This bit is set If the channel 3 input changed from the last comparison with the fixed mux port.
18 CH2F	Channel 2 input changed flag. This bit is set If the channel 2 input changed from the last comparison with the fixed mux port.
17 CH1F	Channel 1 input changed flag. This bit is set If the channel 1 input changed from the last comparison with the fixed mux port.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CMP_C2 field descriptions (continued)

Field	Description
16 CH0F	Channel 0 input changed flag. This bit is set If the channel 0 input changed from the last comparison with the fixed mux port.
15–14 NSAM	Number of sample clocks For a given channel, this field specifies how many round-robin clock cycles later the sample takes place. 00 The comparison result is sampled as soon as the active channel is scanned in one round-robin clock. 01 The sampling takes place 1 round-robin clock cycle after the next cycle of the round-robin clock. 10 The sampling takes place 2 round-robin clock cycles after the next cycle of the round-robin clock. 11 The sampling takes place 3 round-robin clock cycles after the next cycle of the round-robin clock.
13–8 INITMOD	Comparator and DAC initialization delay modulus. These values specify the round robin clock cycles used to determine the comparator and DAC initialization delays specified by the datasheet. For example the initialization delay is 80us and the round robin clock is 100kHz, then INITMOD should be set to 80us/10us = 8. 000000 The modulus is set to 64(same with 111111). other values Inilization delay is set to INITMOD * round robin clock period
ACOn	The result of the input comparison for channel *n* . This field stores the latest comparison result of the input channel *n* with the fixed mux port. Reading this bit returns the latest comparison result. Writing this field defines the pre-set state of channel *n*.

36.9 CMP 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 C0[INVT] = 1.

C0[IER] and C0[IEF] are used to select the condition that causes the CMP module to assert an interrupt to the processor. C0[CFF] is set on a falling edge, and C0[CFR] is set on a rising edge of the comparator output. The optionally filtered CMPO can be read directly through C0[COUT].

36.9.1 Initialization

A typical startup sequence is as follows.

For Assessment Pu

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The time required to stabilize COUT is the power-on delay of the comparators plus the largest propagation delay from a selected analog source through the analog comparator, windowing function, and filter. See the datasheet for power-on delays of the comparators. The windowing function has a maximum of one bus clock period delay. The filter delay is specified in the Low-pass filter section.

During operation, the propagation delay of the selected data paths must always be considered. It may take many bus clock cycles for COUT and C0[CFR]/C0[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 initially equals 0 until sufficient clock cycles have elapsed to fill all stages of the filter. This occurs even if COUTA is at a logic 1.

36.9.2 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 and windowed modes. Filtering can be performed using an internal timebase defined by FPR[FILT_PER], or using an external SAMPLE input 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.

36.9.2.1 Enabling filter modes

Filter modes can be enabled by:

- Setting C0[FILTER_CNT] > 0x01 and
- Setting C0[FPR] to a nonzero value or setting C0[SE]=1

If using the divided bus clock to drive the filter, it samples COUTA every C0[FPR] bus clock cycles.

For Assessment Purposes Only

The filter output is at logic 0 when first initalized, and subsequently changes when all the consecutive C0[FILTER_CNT] samples agree that the output value has changed. In other words, C0[COUT] is 0 for some initial period, even when COUTA is at logic 1.

Setting all of C0[SE], C0[FPR] and C0[FILTER_CNT] 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 C0[FILTER_CNT] on the fly without this intermediate step can result in unexpected behavior.

If C0[SE]=1, the filter samples COUTA on each positive transition of the sample input. The output state of the filter changes when all the consecutive C0[FILTER_CNT] samples agree that the output value has changed.

36.9.2.2 Latency issues

The value of C0[FPR] 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 C0[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 C0[FILTER_CNT].

The values of C0[FPR] or SAMPLE period and C0[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 C0[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 36-4. Comparator sample/filter maximum latencies

Mode #	C0[EN]	C0[WE]	C0[SE]	C0[FILTER_CNT]	Co[FPR]	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		
3A	1	0	1	0x01	X	Sampled, Non-Filtered mode	$T_{PD} + T_{SAMPLE} + T_{per}$

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Freescale Assessment Purposes Only

Table 36-4. Comparator sample/filter maximum latencies (continued)

Mode #	C0[EN]	C0[WE]	C0[SE]	C0[FILTER_CNT]	Co[FPR]	Operation	Maximum latency[1]
3B	1	0	0	0x01	> 0x00		T_{PD} + (C0[FPR] * T_{per}) + T_{per}
4A	1	0	1	> 0x01	X	Sampled, Filtered mode	T_{PD} + (C0[FILTER_CNT] * T_{SAMPLE}) + T_{per}
4B	1	0	0	> 0x01	> 0x00		T_{PD} + (C0[FILTER_CNT] * C0[FPR] x T_{per}) + T_{per}
5A	1	1	0	0x00	X	Windowed mode	T_{PD} + T_{per}
5B	1	1	0	X	0x00		T_{PD} + T_{per}
6	1	1	0	0x01	0x01 - 0xFF	Windowed / Resampled mode	T_{PD} + (C0[FPR] * T_{per}) + $2T_{per}$
7	1	1	0	> 0x01	0x01 - 0xFF	Windowed / Filtered mode	T_{PD} + (C0[FILTER_CNT] * C0[FPR] x T_{per}) + $2T_{per}$

1. T_{PD} represents the intrinsic delay of the analog component plus the polarity select logic. T_{SAMPLE} is the clock period of the external sample clock. T_{per} is the period of the bus clock.

36.10 Interrupts

The CMP module is capable of generating an interrupt on either the rising- or falling-edge of the comparator output, or both. Assuming the CMP DMA enable bit is not set, the following table gives the conditions in which the interrupt request is asserted and deasserted.

Table 36-5. CMP interrupt generations

When	Then
C0[IER] and C0[CFR] are set	The interrupt request is asserted
C0[IEF] and C0[CFF] are set	The interrupt request is asserted
C0[IER] and C0[CFR] are cleared for a rising-edge interrupt	The interrupt request is deasserted
C0[IEF] and C0[CFF] are cleared for a falling-edge interrupt	The interrupt request is deasserted

36.11 DMA support

Normally, the CMP generates a CPU interrupt if there is a change on the COUT. When DMA support is enabled by setting C0[DMAEN] and the interrupt is enabled by setting C0[IER], C0[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

For Assessment Purpose

Freescale Confidential Proprietary - Non-Disclosure Agreement required

sends a transfer completing indicator signal 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.

36.12 DAC functional description

This section provides DAC functional description.

36.12.1 Digital-to-analog converter block diagram

The following figure shows the block diagram of the DAC module. It contains a 256-tap resistor ladder network and a 256-to-1 multiplexer, which selects an output voltage from one of 256 distinct levels that outputs from DACO. It is controlled through the Control register 1 (CMP_C1). 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 the Disabled mode, DACO is connected to the analog ground.

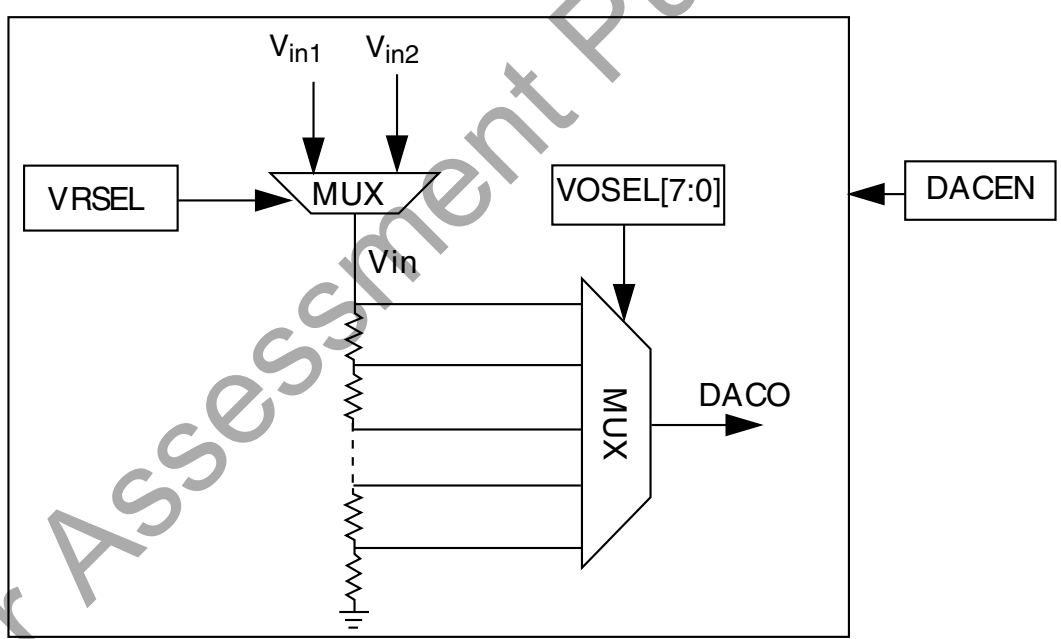


Figure 36-13. 8-bit DAC block diagram

For Assessment Purposes Only

36.12.2 Voltage reference source select

- V_{in1} must be used to connect to the primary voltage source as supply reference of 64 tap resistor ladder
- V_{in2} must be used to connect to an alternate voltage source, or primary source, if an alternate voltage source is not available

36.13 DAC resets

This module has a single reset input, corresponding to the chip-wide peripheral reset.

36.14 DAC clocks

This module has a single clock input, the bus clock.

36.15 DAC interrupts

This module has no interrupts.

36.16 Trigger mode

The CMP and the 8-bit DAC are designed to support the trigger mode operation, which is enabled when the MCU enters STOP modes with CMP_*x*_C2[RRE] and CMP_C0[EN] are set.

With this mode enabled, the trigger events that include the operation clock and a trigger start signal will initiate a compare sequence that must first enable the CMP and DAC prior to performing a CMP operation and capturing the output. A fixed channel for either the plus side mux or the minus side mux is selected by software with CMP_*x*_C2[FXMP] and CMP_*x*_C2[FXMXCH]. It is a mandatory request that the round robin cycling period must be set longer than the time that all the active channels complete the specified comparason cycles specified by CMP_C2[NSAM].

The active channels selected by CMP_*x*_C1[CHN*n*] are then routed to the non-fixed channel mux and compared with the reference input in a round-robin manner. In order to meet the comparator stabilization time, after the configurable number of operation clocks defined by CMP_*x*_C2[NSAM], the comparison result is sampled for the selected

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

channel. A software pre-progammed state for each channel is configured by writing to CMP_*x*_C2[ACO*n*] field. After all the active channels are sampled, if the comparison result changes from its pre-programmed state, the corresponding flag in CMP_*x*_C2[CH*n*F] is set. If CMP_*x*_C2[RRIE] is set, an asynchronous reset is asserted to bring the MCU out of STOP mode.

Note that these flags do not support generating a DMA transfer event.

This mode is active when the MCU is in STOP mode, so none of the window/filter functions are available. A basic assumption of this mode is that the selected inputs are changing at a much slower rate than the operation clock. It is suggested to configure the comparator in low power comparison mode as well. In programming the CMP_C2[INITMOD] registers it is need to make sure the INITMOD*round robin clock period must be longer than the initialization delay which can be referred from the chip datasheet.

The following diagram shows the basic flow of this mode. In the diagram, CMP_*x*_C1[CHN1], CMP_*x*_C1[CHN3], and CMP_*x*_C1[CHN7] are set, so channels #1, #3, and #7 are selected for round robin. CMP_*x*_C2[NSAM] is set to 2'b01, so one clock later the comparison result of the selected channel is sampled. When channel #7 is compared, the result is sampled, and round robin ends. If any of the comparison results from channel #1, #3, or #7 changed from their programmed value (written to CMP_*x*_C2[ACO1], CMP_*x*_C2[ACO3], and CMP_*x*_C2[ACO7]), an interrupt is generated to wake up the MCU from the STOP mode. Software can then poll the CMP_*x*_C2[CH*n*F] to see which channel input(s) changed value during the STOP mode.

NOTE

In round robin mode, it should be ensured that the RTC_CLK period is greater than the comparison time corresponding to the value of CMPx_C0[PMODE]. It is also required to not select the internla reserved channels for round robin by INPSEL and INNSEL.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

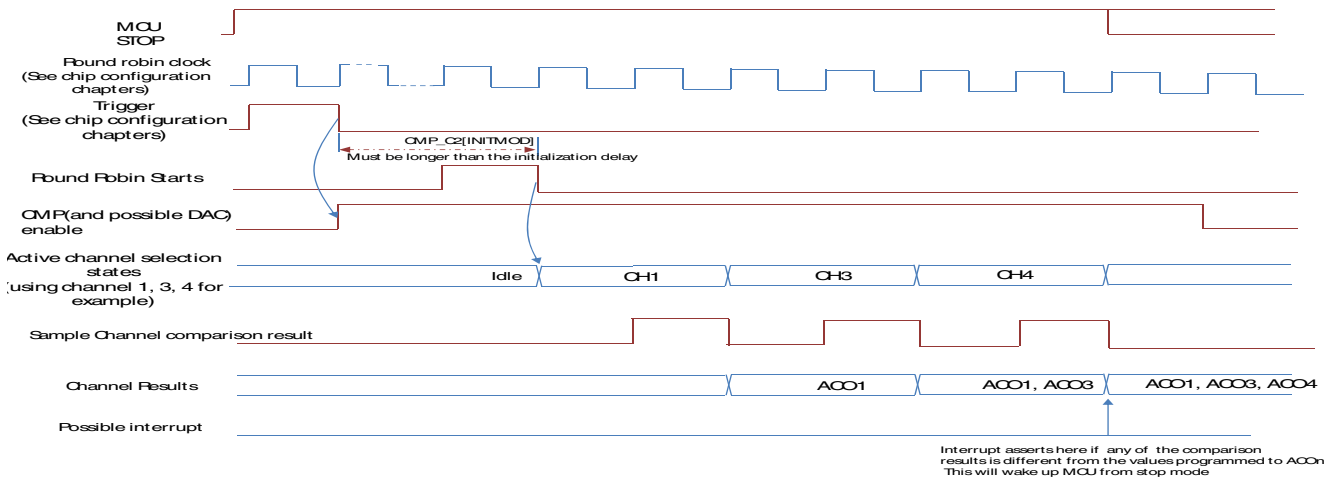


Figure 36-14. Trigger mode

For Assessment Purpose

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 37
Programmable Delay Block (PDB)

37.1 Chip-specific Programmable Delay Block (PDB) information

37.1.1 Instantiation Information

37.1.1.1 PDB Output Triggers

There are 2 PDB modules on this device. Each PDB module has two ADC trigger channels (each ADC channel with 8 pre-triggers), and one Pulse Out channel.

PDB Trigger	PDB output
Number of ADC channel	2
Number of pre-triggers per ADC channel	8
Number of Pulse Out	1

37.1.1.2 PDB Input Trigger Connections

On this device, the PDB trigger source selection is implemented through the TRGMUX module. For each PDB unit, there is only one trigger input from TRGMUX, but it supports different trigger sources. The internal trigger mux inside PDB is not used any more.

PDB Trigger	PDB Input
0000	TRGMUX_PDB_TRIG

For Assessment Purposes Only

37.1.2 PDB Module Interconnections

Besides the specific PDB-ADC triggering scheme (refer to ADC trigger source section), the PDB channel triggers can also work as trigger source of TRGMUX module, which can be used to trigger other modules besides ADC.

Following are PDB module interconnectivities:

PDBx Trigger outputs	Interconnectivity
ADC Channel 0 triggers	ADC trigger connects to both ADC and TRGMUX
ADC Channel 1 trigger	ADC trigger connects to ADC
Pulse-out	Pulse-out connects to TRGMUX

37.1.3 Back-to-back acknowledgement connections

Back-to-back operation enables the ADC conversions complete to trigger the next PDB channel pre-trigger and trigger output.

In this MCU, the following PDB back-to-back operation acknowledgment connections are implemented based on SIM_CHIPCTL[PDB_BB_SEL] bit setting.

When SIM_CHIPCTL[PDB_BB_SEL]=0:

The PDB back-to-back operation acknowledgement connections are implemented inside each PDB unit as a ring.

The back-to-back chain diagram is as follows:

For Assessment Purpose Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

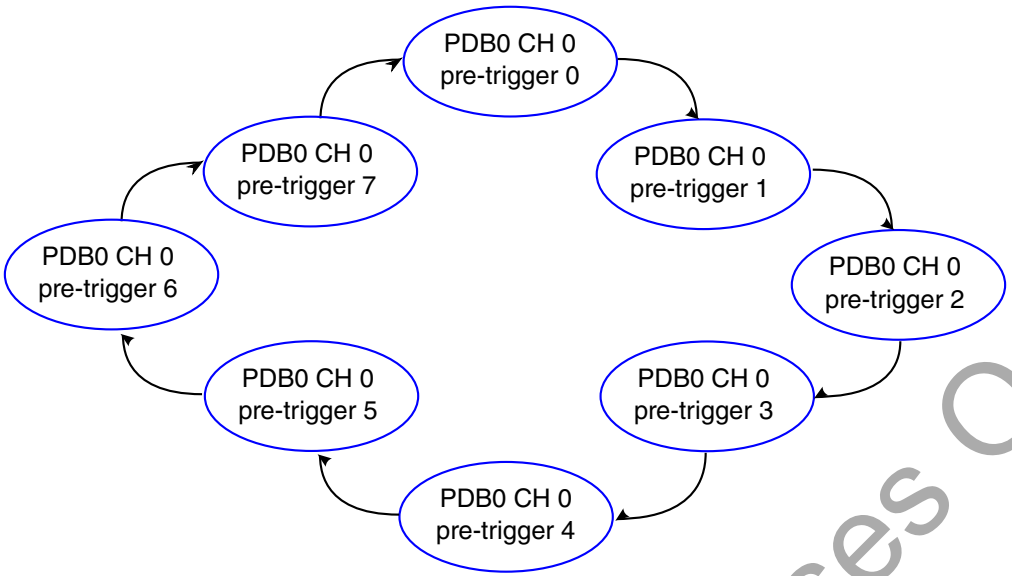


Figure 37-1. PDB0 back-to-back chain

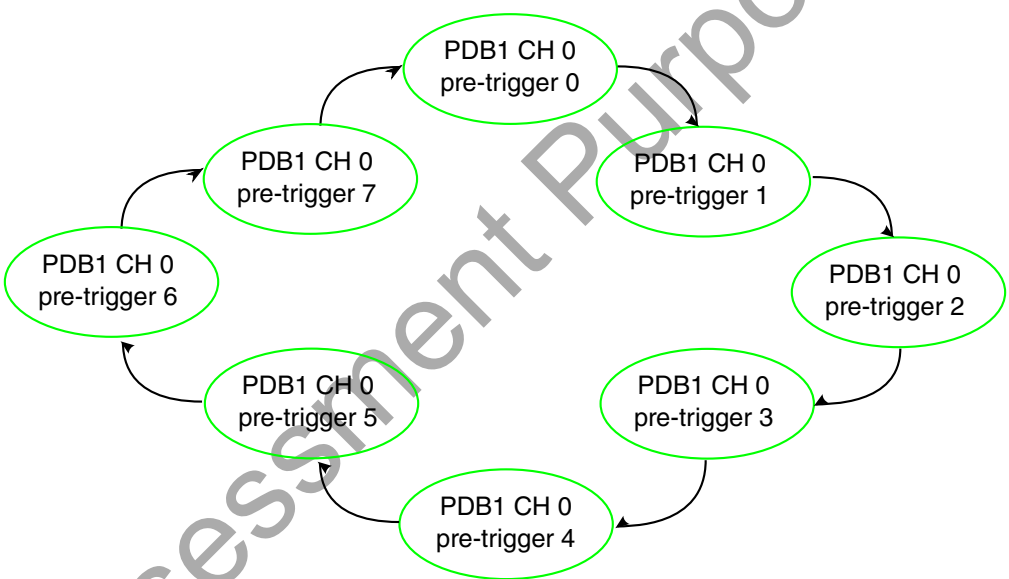


Figure 37-2. PDB1 back-to-back chain

When SIM_CHIPCTL[PDB_BB_SEL]=1:

The PDB back-to-back operation acknowledgement connections are implemented with all of the PDB units as a ring. The chain diagram is as follows:

For Assessment Purposes Only

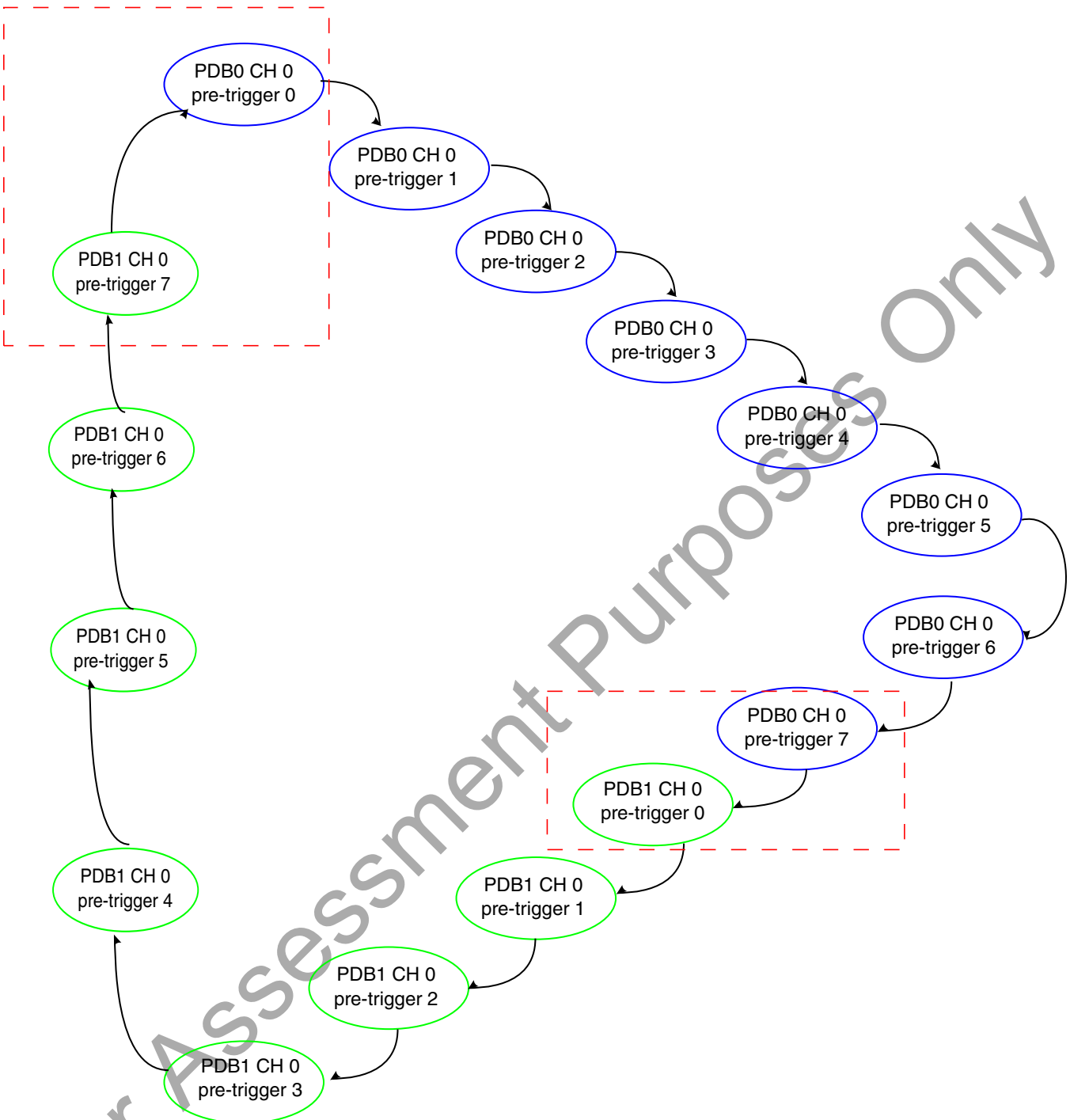


Figure 37-3. PDB back-to-back chain

The application code can set the PDB*x*_CH*n*C1[BB] bits to configure the PDB pre-triggers as a single chain or several chains.

The pre-trigger for the beginning of the chain should be provided from PDB (for e.g., by setting CHnDLY1 and CHnC1[EN] bit) with clearing PDBx_CHnC1[BB] bit.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

37.1.4 Pulse-Out Enable Register Implementation

The following table shows the comparison of pulse-out enable register at the module and chip level. In this device, each PDB unit will have one Pulse-Out channel. The Pulse-Out is connecting to TRGMUX, which is then flexible to work as sample window for any CMP module.

Table 37-1. PDB pulse-out enable register

Register	Module implementation	Chip implementation
POnEN	7:0 - POEN 31:8 - Reserved	0 - POEN[0] for CMP 31:1 - Reserved

37.2 Introduction

The Programmable Delay Block (PDB) provides controllable delays from either an internal or an external trigger, or a programmable interval tick, to the hardware trigger inputs of ADCs. The PDB can optionally provide pulse outputs (Pulse-Out's) that are used as the sample window in the CMP block.

37.2.1 Features

- Up to 2 trigger input sources and one software trigger source
- Up to 8 configurable PDB channels for ADC hardware trigger
  - One PDB channel is associated with one ADC
  - One trigger output for ADC hardware trigger and up to 8 pre-trigger outputs for ADC trigger select per PDB channel
  - Trigger outputs can be enabled or disabled independently
  - One 16-bit delay register per pre-trigger output
  - Optional bypass of the delay registers of the pre-trigger outputs
  - Operation in One-Shot or Continuous modes
  - Optional back-to-back mode operation, which enables the ADC conversions complete to trigger the next PDB channel

For Assessment Purposes Only

- One programmable delay interrupt
- One sequence error interrupt
- One channel flag and one sequence error flag per pre-trigger
- DMA support

- Up to 8 pulse outputs (pulse-out's)
  - Pulse-out's can be enabled or disabled independently
  - Programmable pulse width

NOTE

The number of PDB input and output triggers are chip-specific. See the chip-specific PDB information for details.

37.2.2 Implementation

In this section, the following letters refer to the number of output triggers:

- N—Total available number of PDB channels.
- n—PDB channel number, valid from 0 to *N*-1.
- M—Total available pre-trigger per PDB channel.
- m—Pre-trigger number, valid from 0 to *M*-1.
- Y—Total number of Pulse-Out's.
- y—Pulse-Out number, valid value is from 0 to *Y*-1.

NOTE

The number of module output triggers to core is chip-specific. For module to core output triggers implementation, see the chip configuration information.

37.2.3 Back-to-back acknowledgment connections

PDB back-to-back operation acknowledgment connections are chip-specific. For implementation, see the chip configuration information.

For Assessment Purposes Only

37.2.4 Block diagram

This diagram illustrates the major components of the PDB.

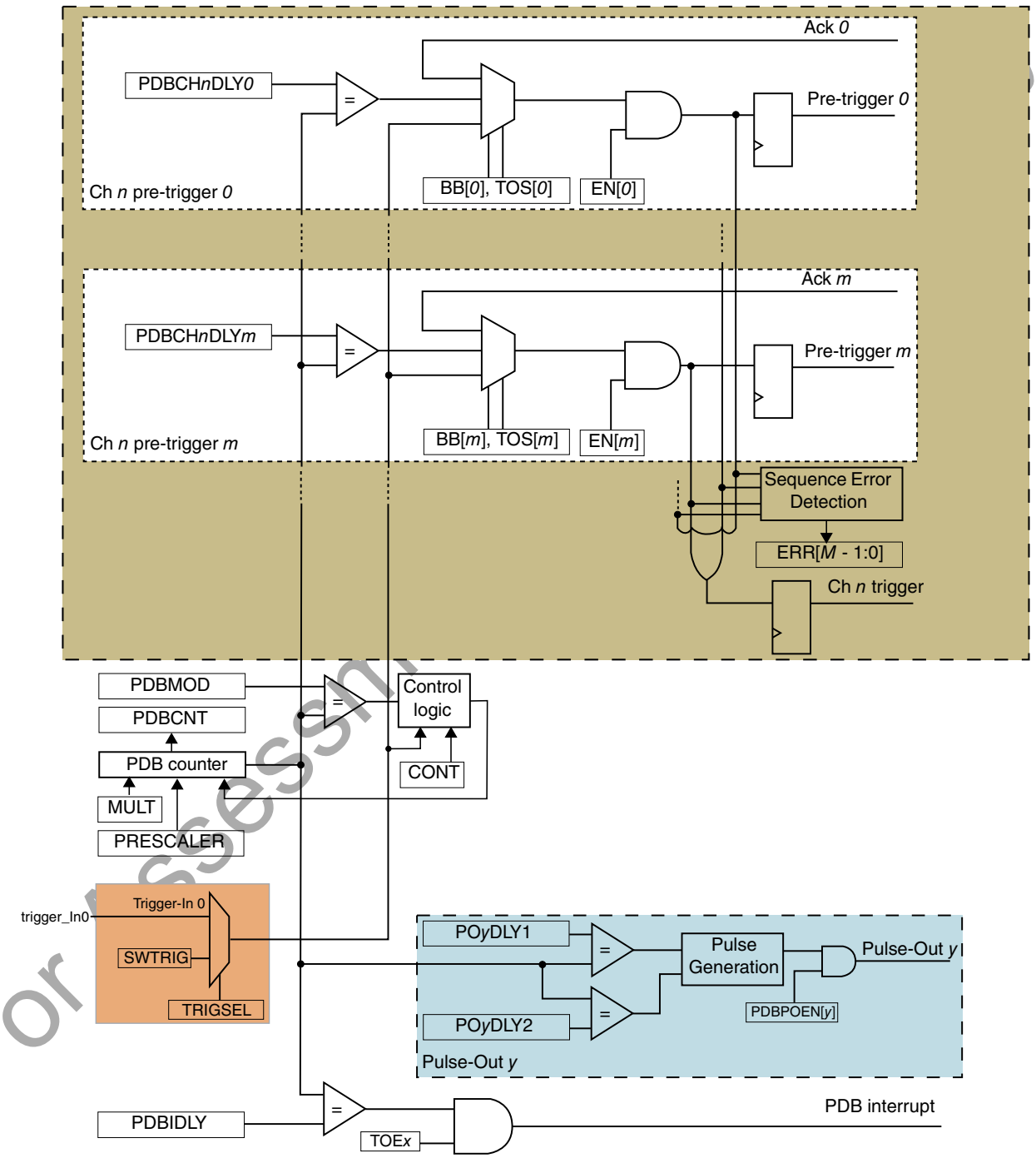


Figure 37-4. PDB block diagram

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

In this diagram, only one PDB channel *n*, and one Pulse-Out *y* are shown. The PDB-enabled control logic and the sequence error interrupt logic are not shown.

37.2.5 Modes of operation

PDB ADC trigger operates in the following modes:

- Disabled—Counter is off, all pre-trigger and trigger outputs are low if PDB is not in back-to-back operation of Bypass mode.
- Debug—Counter is paused when processor is in Debug mode.
- Enabled One-Shot—Counter is enabled and restarted at count zero upon receiving a positive edge on the selected trigger input source or software trigger is selected and SC[SWTRIG] is written with 1. In each PDB channel, an enabled pre-trigger asserts once per trigger input event. The trigger output asserts whenever any of the pre-triggers is asserted.
- Enabled Continuous—Counter is enabled and restarted at count zero. The counter is rolled over to zero again when the count reaches the value specified in the modulus register, and the counting is restarted. This enables a continuous stream of pre-triggers/trigger outputs as a result of a single trigger input event.
- Enabled Bypassed—The pre-trigger and trigger outputs assert immediately after a positive edge on the selected trigger input source or software trigger is selected and SC[SWTRIG] is written with 1, that is the delay registers are bypassed. It is possible to bypass any one or more of the delay registers; therefore, this mode can be used in conjunction with One-Shot or Continuous mode.

37.3 Memory map and register definition

PDB memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Status and Control register (PDB_SC)	32	R/W	0000_0000h	37.3.1/770
4	Modulus register (PDB_MOD)	32	R/W	0000_FFFFh	37.3.2/773
8	Counter register (PDB_CNT)	32	R	0000_0000h	37.3.3/773
C	Interrupt Delay register (PDB_IDLY)	32	R/W	0000_FFFFh	37.3.4/774
10	Channel n Control register 1 (PDB_CH0C1)	32	R/W	0000_0000h	37.3.5/774
14	Channel n Status register (PDB_CH0S)	32	R/W	0000_0000h	37.3.6/775
18	Channel n Delay 0 register (PDB_CH0DLY0)	32	R/W	0000_0000h	37.3.7/776
1C	Channel n Delay 1 register (PDB_CH0DLY1)	32	R/W	0000_0000h	37.3.8/777

Table continues on the next page...

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PDB memory map (continued)

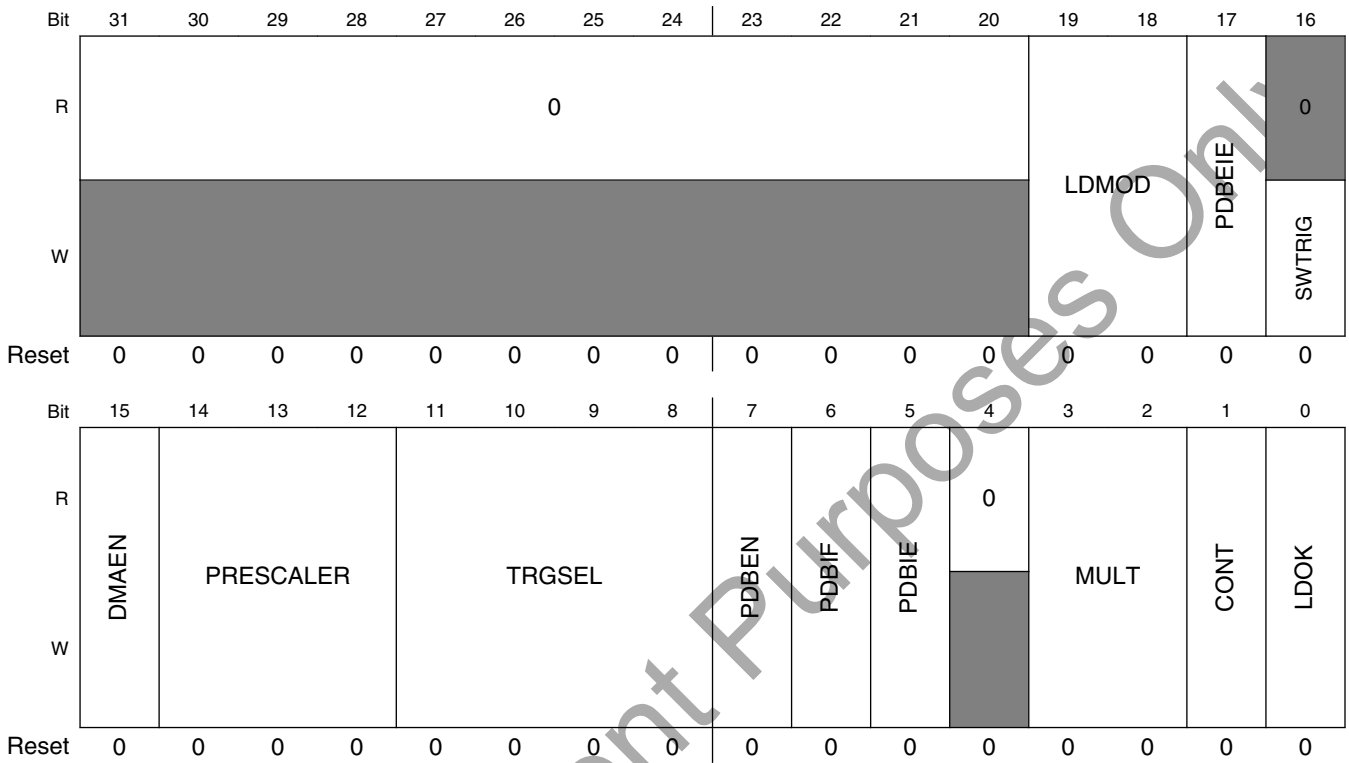
Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20	Channel n Delay 2 register (PDB_CH0DLY2)	32	R/W	0000_0000h	37.3.9/777
24	Channel n Delay 3 register (PDB_CH0DLY3)	32	R/W	0000_0000h	37.3.10/ 778
28	Channel n Delay 4 register (PDB_CH0DLY4)	32	R/W	0000_0000h	37.3.11/ 779
2C	Channel n Delay 5 register (PDB_CH0DLY5)	32	R/W	0000_0000h	37.3.12/ 779
30	Channel n Delay 6 register (PDB_CH0DLY6)	32	R/W	0000_0000h	37.3.13/ 780
34	Channel n Delay 7 register (PDB_CH0DLY7)	32	R/W	0000_0000h	37.3.14/ 781
38	Channel n Control register 1 (PDB_CH1C1)	32	R/W	0000_0000h	37.3.5/774
3C	Channel n Status register (PDB_CH1S)	32	R/W	0000_0000h	37.3.6/775
40	Channel n Delay 0 register (PDB_CH1DLY0)	32	R/W	0000_0000h	37.3.7/776
44	Channel n Delay 1 register (PDB_CH1DLY1)	32	R/W	0000_0000h	37.3.8/777
48	Channel n Delay 2 register (PDB_CH1DLY2)	32	R/W	0000_0000h	37.3.9/777
4C	Channel n Delay 3 register (PDB_CH1DLY3)	32	R/W	0000_0000h	37.3.10/ 778
50	Channel n Delay 4 register (PDB_CH1DLY4)	32	R/W	0000_0000h	37.3.11/ 779
54	Channel n Delay 5 register (PDB_CH1DLY5)	32	R/W	0000_0000h	37.3.12/ 779
58	Channel n Delay 6 register (PDB_CH1DLY6)	32	R/W	0000_0000h	37.3.13/ 780
5C	Channel n Delay 7 register (PDB_CH1DLY7)	32	R/W	0000_0000h	37.3.14/ 781
190	Pulse-Out n Enable register (PDB_POEN)	32	R/W	0000_0000h	37.3.15/ 781
194	Pulse-Out n Delay register (PDB_PO0DLY)	32	R/W	0000_0000h	37.3.16/ 782

For Ass

Freescale Confidential Proprietary - Non-Disclosure Agreement required

37.3.1 Status and Control register (PDB_SC)

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



Purposes Only

PDB_SC field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–18 LDMOD	Load Mode Select Selects the mode to load the MOD, IDLY, CH*n*DLY*m*, INT*x*, and PO*y*DLY registers, after 1 is written to LDOK. 00 The internal registers are loaded with the values from their buffers immediately after 1 is written to LDOK. 01 The internal registers are loaded with the values from their buffers when the PDB counter reaches the MOD register value after 1 is written to LDOK. 10 The internal registers are loaded with the values from their buffers when a trigger input event is detected after 1 is written to LDOK. 11 The internal registers are loaded with the values from their buffers when either the PDB counter reaches the MOD register value or a trigger input event is detected, after 1 is written to LDOK.
17 PDBEIE	PDB Sequence Error Interrupt Enable Enables the PDB sequence error interrupt. When this field is set, any of the PDB channel sequence error flags generates a PDB sequence error interrupt.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PDB_SC field descriptions (continued)

Field	Description
	0 PDB sequence error interrupt disabled. 1 PDB sequence error interrupt enabled.
16 SWTRIG	Software Trigger When PDB is enabled and the software trigger is selected as the trigger input source, writing 1 to this field resets and restarts the counter. Writing 0 to this field has no effect. Reading this field results 0.
15 DMAEN	DMA Enable When DMA is enabled, the PDBIF flag generates a DMA request instead of an interrupt. 0 DMA disabled. 1 DMA enabled.
14–12 PRESCALER	Prescaler Divider Select 000 Counting uses the peripheral clock divided by multiplication factor selected by MULT. 001 Counting uses the peripheral clock divided by twice of the multiplication factor selected by MULT. 010 Counting uses the peripheral clock divided by four times of the multiplication factor selected by MULT. 011 Counting uses the peripheral clock divided by eight times of the multiplication factor selected by MULT. 100 Counting uses the peripheral clock divided by 16 times of the multiplication factor selected by MULT. 101 Counting uses the peripheral clock divided by 32 times of the multiplication factor selected by MULT. 110 Counting uses the peripheral clock divided by 64 times of the multiplication factor selected by MULT. 111 Counting uses the peripheral clock divided by 128 times of the multiplication factor selected by MULT.
11–8 TRGSEL	Trigger Input Source Select Selects the trigger input source for the PDB. The trigger input source can be internal or external (EXTRG pin), or the software trigger. Refer to chip configuration details for the actual PDB input trigger connections. 0000 Trigger-In 0 is selected. 0001 Trigger-In 1 is selected. 0010 Trigger-In 2 is selected. 0011 Trigger-In 3 is selected. 0100 Trigger-In 4 is selected. 0101 Trigger-In 5 is selected. 0110 Trigger-In 6 is selected. 0111 Trigger-In 7 is selected. 1000 Trigger-In 8 is selected. 1001 Trigger-In 9 is selected. 1010 Trigger-In 10 is selected. 1011 Trigger-In 11 is selected. 1100 Trigger-In 12 is selected. 1101 Trigger-In 13 is selected. 1110 Trigger-In 14 is selected. 1111 Software trigger is selected.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PDB_SC field descriptions (continued)

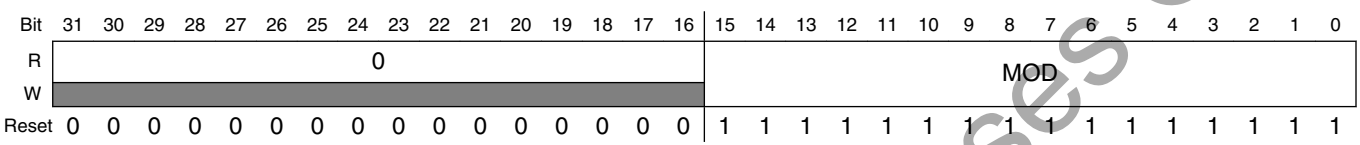
Field	Description
7 PDBEN	PDB Enable 0 PDB disabled. Counter is off. 1 PDB enabled.
6 PDBIF	PDB Interrupt Flag This field is set when the counter value is equal to the IDLY register. Writing zero clears this field.
5 PDBIE	PDB Interrupt Enable Enables the PDB interrupt. When this field is set and DMAEN is cleared, PDBIF generates a PDB interrupt. 0 PDB interrupt disabled. 1 PDB interrupt enabled.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–2 MULT	Multiplication Factor Select for Prescaler Selects the multiplication factor of the prescaler divider for the counter clock. 00 Multiplication factor is 1. 01 Multiplication factor is 10. 10 Multiplication factor is 20. 11 Multiplication factor is 40.
1 CONT	Continuous Mode Enable Enables the PDB operation in Continuous mode. 0 PDB operation in One-Shot mode 1 PDB operation in Continuous mode
0 LDOK	Load OK Writing 1 to LDOK bit updates the MOD, IDLY, CHnDLYm, and POyDLY registers with the values previously written to their internal buffers (and stored there). The new values of MOD, IDLY, CHnDLYm, and POyDLY registers will take effect according to the setting of the LDMOD field (Load Mode Select). After 1 is written to the LDOK field, the values in the internal buffers of these registers are not effective, and new values cannot be written to the internal buffers until the existing values in the internal buffers are loaded into their corresponding registers. • LDOK can be written only when PDBEN is set, or LDOK can be written at the same time when PDBEN is written to 1. • LDOK is automatically cleared when the values in the internal buffers are loaded into the registers or when PDBEN bit (PDB Enable) is cleared. • Writing 0 to LDOK has no effect.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

37.3.2 Modulus register (PDB_MOD)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + 4h offset = 4h

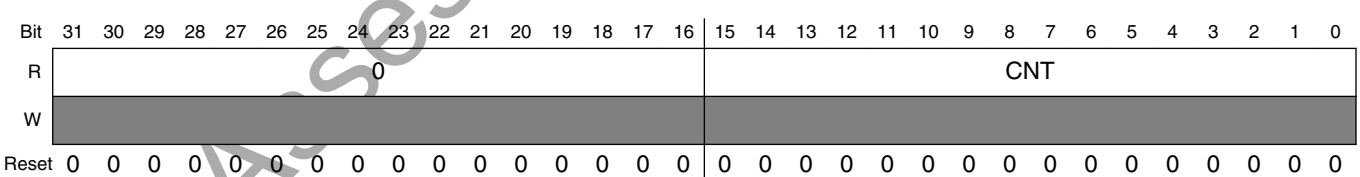


PDB_MOD field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MOD	PDB Modulus Specifies the period of the counter. When the counter reaches this value, it will be reset back to zero. If the PDB is in Continuous mode, the count begins anew. Reading this field returns the value of the internal register that is effective for the current cycle of PDB.

37.3.3 Counter register (PDB_CNT)

Address: 0h base + 8h offset = 8h



PDB_CNT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CNT	PDB Counter Contains the current value of the counter.

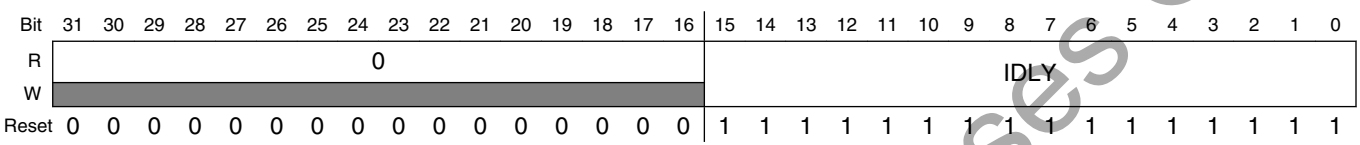
Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

37.3.4 Interrupt Delay register (PDB_IDLY)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + Ch offset = Ch



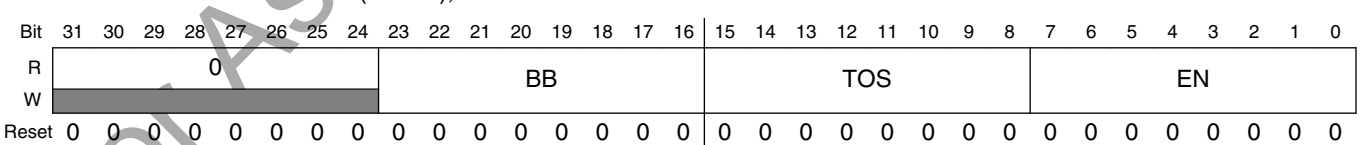
PDB_IDLY field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
IDLY	PDB Interrupt Delay Specifies the delay value to schedule the PDB interrupt. It can be used to schedule an independent interrupt at some point in the PDB cycle. If enabled, a PDB interrupt is generated, when the counter is equal to the IDLY. Reading this field returns the value of internal register that is effective for the current cycle of the PDB.

37.3.5 Channel n Control register 1 (PDB_CH*n*C1)

Each PDB channel has one control register, CHnC1. The bits in this register control the functionality of each PDB channel operation.

Address: 0h base + 10h offset + (40d × i), where i=0d to 1d



PDB_CH*n*C1 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 BB	PDB Channel Pre-Trigger Back-to-Back Operation Enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PDB_CH*n*C1 field descriptions (continued)

Field	Description
	These bits enable the PDB ADC pre-trigger operation as back-to-back mode. Only lower M pre-trigger bits are implemented in this MCU. Back-to-back operation enables the ADC conversions complete to trigger the next PDB channel pre-trigger and trigger output, so that the ADC conversions can be triggered on next set of configuration and results registers. Application code must only enable the back-to-back operation of the PDB pre-triggers at the leading of the back-to-back connection chain. 0 PDB channel's corresponding pre-trigger back-to-back operation disabled. 1 PDB channel's corresponding pre-trigger back-to-back operation enabled.
15–8 TOS	PDB Channel Pre-Trigger Output Select Selects the PDB ADC pre-trigger outputs. Only lower M pre-trigger fields are implemented in this MCU. 0 PDB channel's corresponding pre-trigger is in bypassed mode. The pre-trigger asserts one peripheral clock cycle after a rising edge is detected on selected trigger input source or software trigger is selected and SWTRIG is written with 1. 1 PDB channel's corresponding pre-trigger asserts when the counter reaches the channel delay register and one peripheral clock cycle after a rising edge is detected on selected trigger input source or software trigger is selected and SETRIG is written with 1.
EN	PDB Channel Pre-Trigger Enable These bits enable the PDB ADC pre-trigger outputs. Only lower M pre-trigger bits are implemented in this MCU. 0 PDB channel's corresponding pre-trigger disabled. 1 PDB channel's corresponding pre-trigger enabled.

Assessment

37.3.6 Channel n Status register (PDB_CH*n*S)

Address: 0h base + 14h offset + (40d × 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								CF								0								ERR							
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

PDB_CH*n*S field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 CF	PDB Channel Flags The CF[*m*] bit is set when the PDB counter matches the CH*n*DLY*m*. Write 0 to clear these bits.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ERR	PDB Channel Sequence Error Flags Only the lower M bits are implemented in this MCU.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PDB_CH*n*S field descriptions (continued)

Field	Description
	0 Sequence error not detected on PDB channel's corresponding pre-trigger. 1 Sequence error detected on PDB channel's corresponding pre-trigger. ADCn block can be triggered for a conversion by one pre-trigger from PDB channel *n*. When one conversion, which is triggered by one of the pre-triggers from PDB channel *n*, is in progress, new trigger from PDB channel's corresponding pre-trigger m cannot be accepted by ADCn, and ERR[m] is set. Writing 0's to clear the sequence error flags.

37.3.7 Channel n Delay 0 register (PDB_CH*n*DLY0)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + 18h offset + (40d × 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																DLY															
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

PDB_CH*n*DLY0 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DLY	PDB Channel Delay Specifies the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading this field returns the value of internal register that is effective for the current PDB cycle.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

37.3.8 Channel n Delay 1 register (PDB_CH*n*DLY1)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + 1Ch offset + (40d × 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																DLY															
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

PDB_CH*n*DLY1 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

37.3.9 Channel n Delay 2 register (PDB_CH*n*DLY2)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + 20h offset + (40d × 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																DLY															
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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PDB_CH*n*DLY2 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

37.3.10 Channel n Delay 3 register (PDB_CH*n*DLY3)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + 24h offset + (40d × 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																DLY															
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

PDB_CH*n*DLY3 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

37.3.11 Channel n Delay 4 register (PDB_CH*n*DLY4)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + 28h offset + (40d × 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																DLY															
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

PDB_CH*n*DLY4 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

37.3.12 Channel n Delay 5 register (PDB_CH*n*DLY5)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + 2Ch offset + (40d × 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																DLY															
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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

PDB_CH*n*DLY5 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

37.3.13 Channel n Delay 6 register (PDB_CH*n*DLY6)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + 30h offset + (40d × 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																DLY															
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

PDB_CH*n*DLY6 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

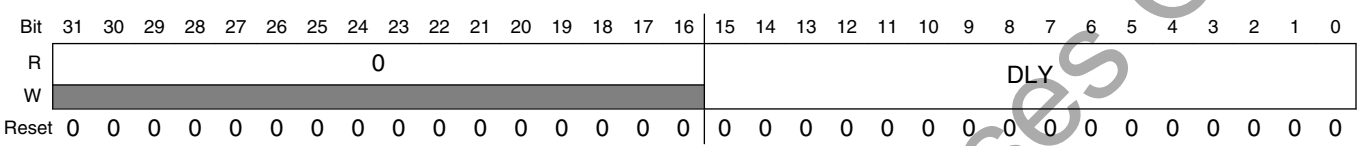
For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

37.3.14 Channel n Delay 7 register (PDB_CH*n*DLY7)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + 34h offset + (40d × i), where i=0d to 1d

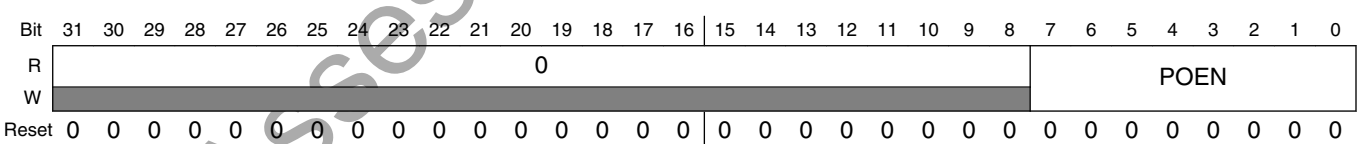


PDB_CH*n*DLY7 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

37.3.15 Pulse-Out n Enable register (PDB_POEN)

Address: 0h base + 190h offset = 190h



PDB_POEN field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
POEN	PDB Pulse-Out Enable Enables the pulse output. Only lower Y bits are implemented in this MCU. 0 PDB Pulse-Out disabled 1 PDB Pulse-Out enabled

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

37.3.16 Pulse-Out n Delay register (PDB_PO*n*DLY)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 0h base + 194h offset + (4d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	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	DLY1																DLY2															
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

PDB_PO*n*DLY field descriptions

Field	Description
31–16 DLY1	PDB Pulse-Out Delay 1 These bits specify the delay 1 value for the PDB Pulse-Out. Pulse-Out goes high when the PDB counter is equal to the DLY1. Reading these bits returns the value of internal register that is effective for the current PDB cycle.
DLY2	PDB Pulse-Out Delay 2 These bits specify the delay 2 value for the PDB Pulse-Out. Pulse-Out goes low when the PDB counter is equal to the DLY2. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

37.4 Functional description

37.4.1 PDB pre-trigger and trigger outputs

The PDB contains a counter whose output is compared to several different digital values. If the PDB is enabled, then a trigger input event will reset the counter and make it start to count. A trigger input event is defined as a rising edge being detected on a selected trigger input source, or if a software trigger is selected and SC[SWTRIG] is written with 1. For each channel, a delay *m* determines the time between assertion of the trigger input event to the time at which changes in the pre-trigger *m* output signal are started. The time is defined as:

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- Trigger input event to pre-trigger *m* = (prescaler × multiplication factor × delay *m*) + 2 peripheral clock cycles
- Add 1 additional peripheral clock cycle to determine the time when the channel trigger output changes.

Each channel is associated with 1 ADC block. PDB channel n pre-trigger outputs 0 to *M*; each pre-trigger output is connected to ADC hardware trigger select and hardware trigger inputs. The pre-triggers are used to precondition the ADC block before the actual trigger occurs. When the ADC receives the rising edge of the trigger, the ADC will start the conversion according to the precondition determined by the pre-triggers. The ADC contains *M* sets of configuration and result registers, allowing it to alternate conversions between *M* different analog sources (like a ping-pong game). The pre-trigger outputs are used to specify which signal will be sampled next. When a pre-trigger *m* is asserted, the ADC conversion is triggered with set *m* of the configuration and result registers.

The waveforms shown in the following diagram show the pre-trigger and trigger outputs of PDB channel *n*. The delays can be independently set using the CH*n*DLY*m* registers, and the pre-triggers can be enabled or disabled in CH*n*C1[EN[*m*]].

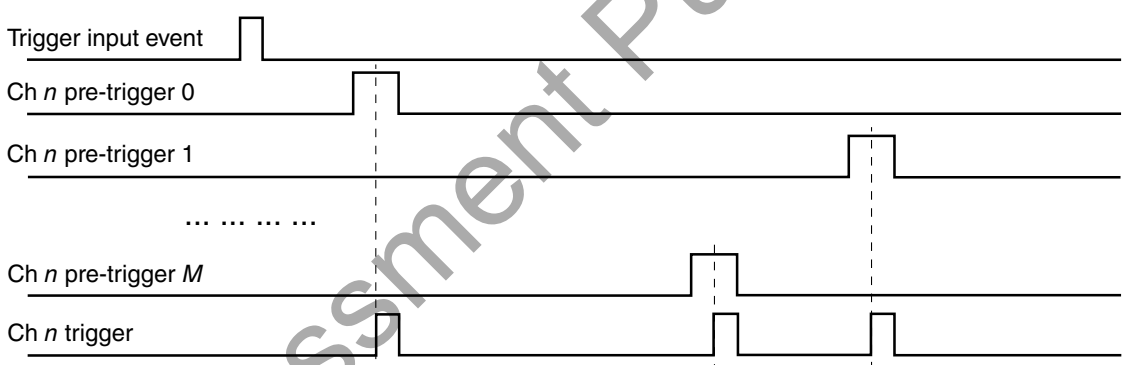


Figure 37-5. Pre-trigger and trigger outputs

The delay in CH*n*DLY*m* register can be optionally bypassed, if CH*n*C1[TOS[*m*]] is cleared. In this case, when the trigger input event occurs, the pre-trigger *m* is asserted after 2 peripheral clock cycles.

The PDB can be configured for back-to-back operation. Back-to-back operation enables the ADC conversion completions to trigger the next PDB channel pre-trigger and trigger outputs, so that the ADC conversions can be triggered on the next set of configuration and results registers. When back-to-back operation is enabled by setting CH*n*C1[BB[*m*]], then the delay *m* is ignored and the pre-trigger *m* is asserted 2 peripheral cycles after the acknowledgment *m* is received. The acknowledgment connections in this MCU are described in Back-to-back acknowledgment connections.

For Assessment Purposes Only

When a pre-trigger from a PDB channel *n* is asserted, the associated lock of the pre-trigger becomes active. The associated lock is released by the rising edge of the corresponding ADC*n*SC1[COCO]; the ADC*n*SC1[COCO] should be cleared after the conversion result is read, so that the next rising edge of ADC*n*SC1[COCO] can be generated to clear the lock later. The lock becomes inactive when:

- the rising edge of corresponding ADC*n*SC1[COCO] occurs,
- or the corresponding PDB pre-trigger is disabled,
- or the PDB is disabled

The channel *n* trigger output is suppressed when any of the locks of the pre-triggers in channel *n* is active. If a new pre-trigger *m* asserts when there is active lock in the PDB channel *n*, then a register flag bit CH*n*S[ERR[*m*]] (associated with the pre-trigger *m*) is set. If SC[PDBEIE] is set, then the sequence error interrupt is generated. A sequence error typically happens because the delay *m* is set too short and the pre-trigger *m* asserts before the previously triggered ADC conversion finishes.

When the PDB counter reaches the value set in IDLY register, the SC[PDBIF] flag is set. A PDB interrupt can be generated if SC[PDBIE] is set and SC[DMAEN] is cleared. If SC[DMAEN] is set, then the PDB requests a DMA transfer when the SC[PDBIF] flag is set.

The modulus value in the MOD register is used to reset the counter back to zero at the end of the count. If SC[CONT] is set, then the counter will then resume a new count; otherwise, the counter operation will stop until the next trigger input event occurs.

37.4.2 PDB trigger input source selection

The PDB has up to 3 trigger input sources: software trigger, internal trigger, external trigger (via a pin). They are connected to on-chip or off-chip event sources. The PDB can be triggered by software through SC[SWTRIG].

For the trigger input sources implemented in this MCU, see chip configuration information.

37.4.3 Pulse-Out's

PDB can generate pulse outputs of configurable width.

- When the PDB counter reaches the value set in PO*y*DLY[DLY1], then the Pulse-Out goes high.
- When the PDB counter reaches PO*y*DLY[DLY2], then it goes low.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

POyDLY[DLY2] can be set either greater or less than POyDLY[DLY1].

ADC pre-trigger/trigger outputs and Pulse-Out generation have the same time base, because they both share the PDB counter. The pulse-out connections implemented in this MCU are described in the device's chip configuration details.

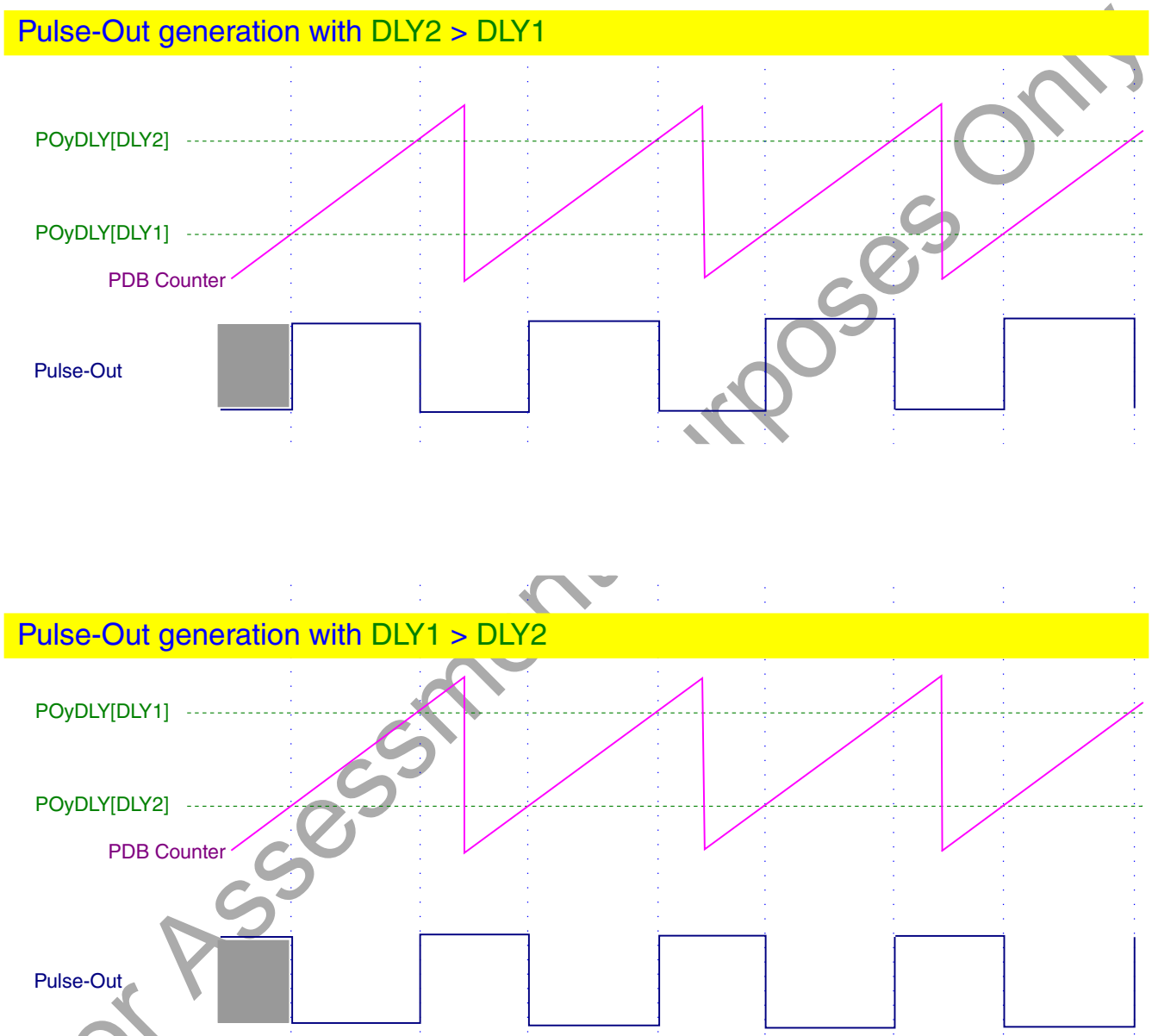


Figure 37-6. How Pulse Out is generated

For Assessment Purposes Only

37.4.4 Updating the delay registers

The following registers control the timing of the PDB operation; and in some of the applications, they may need to become effective at the same time.

- PDB Modulus register (MOD)
- PDB Interrupt Delay register (IDLY)
- PDB Channel *n* Delay *m* register (CH*n*DLY*m*)
- PDB Pulse-Out *y* Delay register (PO*y*DLY)

The internal registers of them are buffered and any values written to them are written first to their buffers. The circumstances that cause their internal registers to be updated with the values from the buffers are summarized as shown in the table below.

Table 37-2. Circumstances of update to the delay registers

SC[LDMOD]	Update to the delay registers
00	The internal registers are loaded with the values from their buffers immediately after 1 is written to SC[LDOK].
01	The PDB counter reaches the MOD register value after 1 is written to SC[LDOK].
10	A trigger input event is detected after 1 is written to SC[LDOK].
11	Either the PDB counter reaches the MOD register value, or a trigger input event is detected, after 1 is written to SC[LDOK].

After 1 is written to SC[LDOK], the buffers cannot be written until the values in buffers are loaded into their internal registers. SC[LDOK] is self-cleared when the internal registers are loaded, so the application code can read it to determine the updates to the internal registers.

The following diagrams show the cases of the internal registers being updated with SC[LDMOD] is 00 and x1.

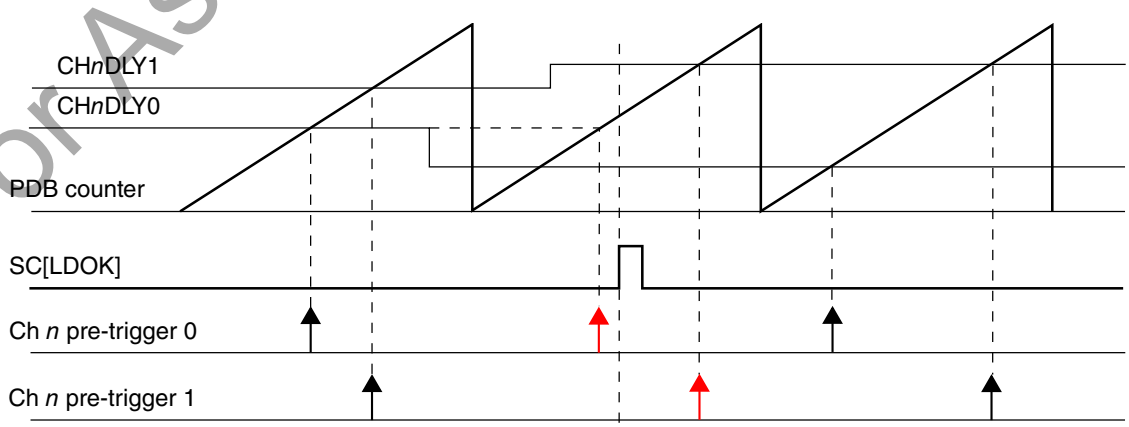


Figure 37-7. Registers update with SC[LDMOD] = 00

For Assessment Purposes Only

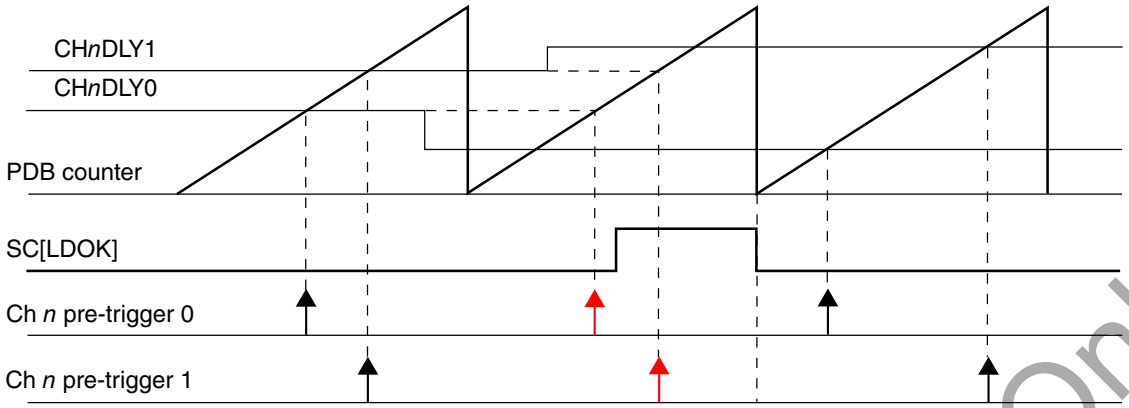


Figure 37-8. Registers update with SC[LDMOD] = x1

37.4.5 Interrupts

PDB can generate two interrupts: PDB interrupt and PDB sequence error interrupt. The following table summarizes the interrupts.

Table 37-3. PDB interrupt summary

Interrupt	Flags	Enable bit
PDB Interrupt	SC[PDBIF]	SC[PDBIE] = 1 and SC[DMAEN] = 0
PDB Sequence Error Interrupt	CH*n*S[ERR*m*]	SC[PDBEIE] = 1

37.4.6 DMA

If SC[DMAEN] is set, PDB can generate a DMA transfer request when SC[PDBIF] is set. When DMA is enabled, the PDB interrupt is not issued.

37.5 Application information

37.5.1 Impact of using the prescaler and multiplication factor on timing resolution

Use of prescaler and multiplication factor greater than 1 limits the count/delay accuracy in terms of peripheral clock cycles (to the modulus of the prescaler X multiplication factor). If the multiplication factor is set to 1 and the prescaler is set to 2 then the only

For Assessment Purposes Only

values of total peripheral clocks that can be detected are even values; if prescaler is set to 4 then the only values of total peripheral clocks that can be decoded as detected are mod(4) and so forth. If the applications need a really long delay value and use a prescaler set to 128, then the resolution would be limited to 128 peripheral clock cycles.

Therefore, use the lowest possible prescaler and multiplication factor for a given application.

For Assessment Purposes Only

Chapter 38
FlexTimer Module (FTM)

38.1 Chip-specific FlexTimer Module (FTM) information

38.1.1 Instantiation Information

This device contains four FlexTimer modules.

The following table shows how these modules are configured.

Table 38-1. FTM Instantiations

FTM instance	Number of channels	Features/usage
FTM0	8	FTM enhanced features, GTB_EN
FTM1	8	FTM enhanced features, GTB_EN, Quadrature Decoder
FTM2	8	FTM enhanced features, GTB_EN, Quadrature Decoder
FTM3	8	FTM enhanced features, GTB_EN

Compared with the FTM0 and FTM3 configuration, the FTM1 and FTM2 configuration adds the Quadrature decoder feature.

38.1.2 FTM Interrupts

The FlexTimer has multiple sources of interrupt. Refer to the DMA_Interrupt_mapping.xlsm attached with this Reference Manual. When an FTM interrupt occurs, read the FTM status registers (FMS, SC, and STATUS) to determine the exact interrupt source.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.1.3 FTM Fault Detection Inputs

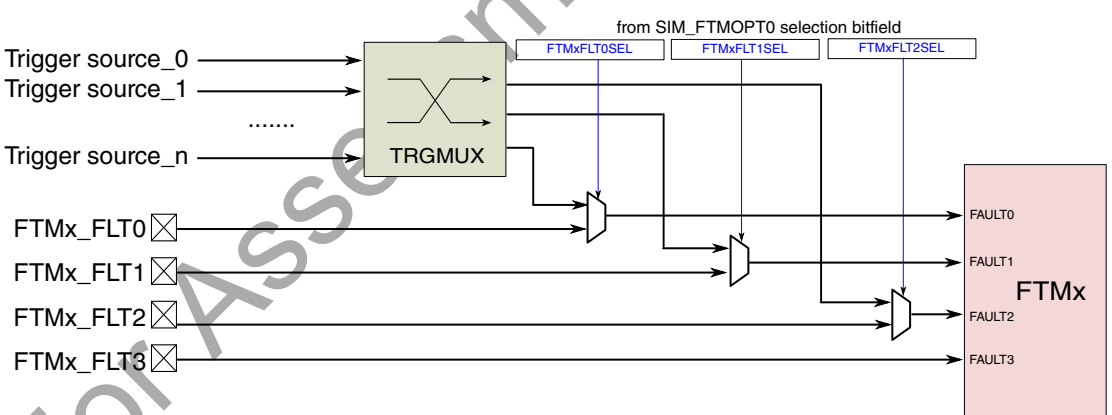
The following fault detection input options for the FTM modules are selected via the SIM_FTMOPT0 register. The external pin option is selected by default.

- FTM0 FAULT0 = FTM0_FLT0 pin or TRGMUX output
- FTM0 FAULT1 = FTM0_FLT1 pin or TRGMUX output
- FTM0 FAULT2 = FTM0_FLT2 pin or TRGMUX output
- FTM0 FAULT3 = FTM0_FLT3 pin

- FTM1 FAULT0 = FTM1_FLT0 pin or TRGMUX output
- FTM1 FAULT1 = FTM1_FLT1 pin or TRGMUX output
- FTM1 FAULT2 = FTM1_FLT2 pin or TRGMUX output
- FTM1 FAULT3 = FTM1_FLT3 pin

- FTM2 FAULT0 = FTM2_FLT0 pin or TRGMUX output
- FTM2 FAULT1 = FTM2_FLT1 pin or TRGMUX output
- FTM2 FAULT2 = FTM2_FLT2 pin or TRGMUX output
- FTM2 FAULT3 = FTM2_FLT3 pin

- FTM3 FAULT0 = FTM3_FLT0 pin or TRGMUX output
- FTM3 FAULT1 = FTM3_FLT1 pin or TRGMUX output
- FTM3 FAULT2 = FTM3_FLT2 pin or TRGMUX output
- FTM3 FAULT3 = FTM3_FLT3 pin



38.1.4 FTM Hardware Triggers and Synchronization

The FlexTimer support external hardware trigger input which can be used for timer dynamic synchronization between multiple FlexTimers or counter reset. The FlexTimer hardware trigger are implemented as following.

For Assessment Purposes Only

FTM0:

- FTM0 hardware trigger 0 = TRGMUX trigger output
- FTM0 hardware trigger 1 = SIM_FTMOPT1[FTM0SYNCBIT]
- FTM0 hardware trigger 2 = FTM0_FLT0 pin

FTM1:

- FTM1 hardware trigger 0 = TRGMUX trigger output
- FTM1 hardware trigger 1 = SIM_FTMOPT1[FTM1SYNCBIT]
- FTM1 hardware trigger 2 = FTM1_FLT0 pin

FTM2:

- FTM2 hardware trigger 0 = TRGMUX trigger output
- FTM2 hardware trigger 1 = SIM_FTMOPT1[FTM2SYNCBIT]
- FTM2 hardware trigger 2 = FTM2_FLT0 pin

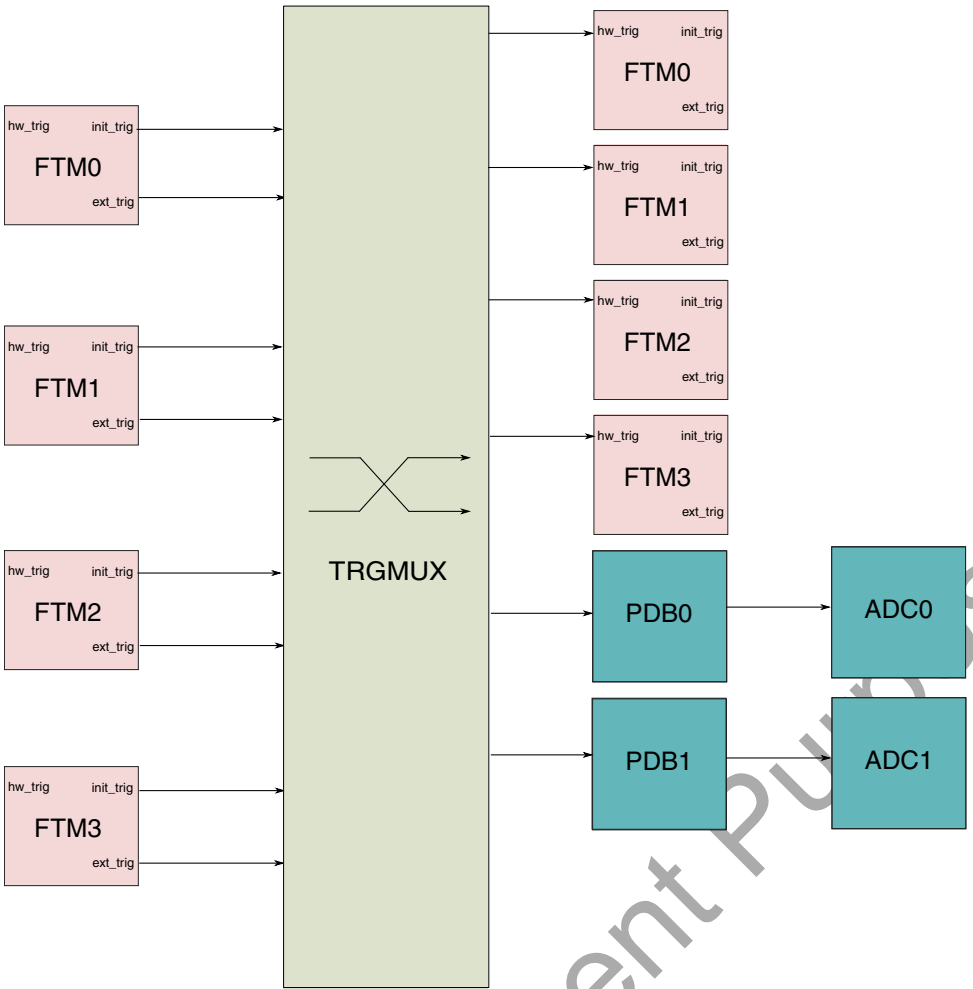
FTM3:

- FTM3 hardware trigger 0 = TRGMUX trigger output
- FTM3 hardware trigger 1 = SIM_FTMOPT1[FTM3SYNCBIT]
- FTM3 hardware trigger 2 = FTM3_FLT0 pin

The hardware trigger source can be from many other modules via TRGMUX, like LPIT, Low Power Timer, CMP, etc. It also supports FlexTimer's self trigger outputs, ex: counter initialization trigger (init_trig) and channel match trigger (ext_trig), through the flexible TRGMUX module.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required



The FlexTimer trigger outputs are also usually used as trigger source by other modules, for example, the above diagram shows a case of triggering PDB and ADC. See Instantiation Information and ADC triggers sections for details.

38.1.5 FTM Input Capture Options

The following channel 0 input capture source options are selected via SIM_FTMOPT0. The external pin option is selected by default.

- FTM1 channel 0 input capture = FTM1_CH0 pin or CMP0 output
- FTM2 channel 0 input capture = FTM2_CH0 pin or CMP0 output
- FTM2 channel 1 input capture = FTM2_CH1 pin or exclusive OR of FTM2_CH0, FTM2_CH1, and FTM1_CH1. See FTM Hall sensor support.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.1.6 FTM Hall sensor support

For 3 phase motor control sensor-ed applications the use of Hall sensors, generally 3 sensors placed 120 degrees apart around the rotor, are deployed to detect position and speed. Each of the 3 sensors provides a pulse that applied to an input capture pin, can then be analyzed and both speed and position can be deduced. To simplify the calculations required by the CPU on each hall sensor's input, if all 3 inputs are "exclusively OR'd" into one timer channel and the free running counter is refreshed on every edge then this can simplify the speed calculation.

Via the SIM module and SIM_FTMOPT1 register the FTM2CH1SEL bit provides the choice of normal FTM2_CH1 input or the XOR of FTM2_CH0, FTM2_CH1 and FTM1_CH1 pins that will be applied to FTM2_CH1.

NOTE

If the user utilizes FTM1_CH1 to be an input to FTM2_CH1, FTM1_CH0 can still be utilized for other functions.

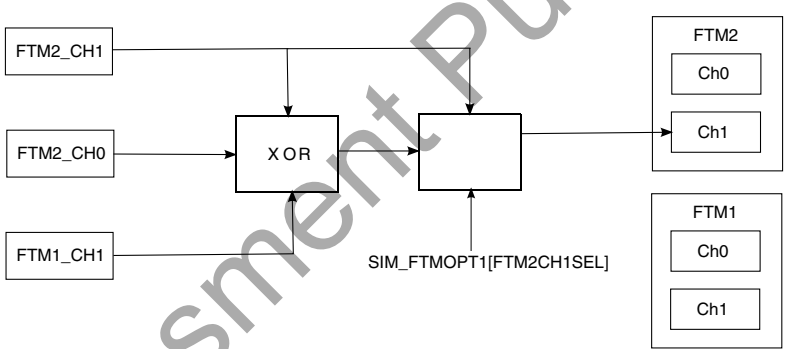


Figure 38-1. FTM Hall Sensor Configuration

38.1.7 FTM Modulation Implementation

FTM0 and FTM3 support a modulation function where the output channels when configured as PWM or Output Compare mode modulate another timer output when the channel signal is asserted. Any of the 8 channels of FTM0 and any of the 8 channels of FTM3 can be configured to support this modulation function.

The SIM_FTMOPT1 register has control bits (FTMxCHySEL) that allow the user to select normal PWM/Output Compare mode on the corresponding FTM timer channel or modulate with FTM1_CH1. The diagram below shows the implementation for FTM0.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM3 has similar implementation controlled by SIM_FTMOPT1[FTM3CHySEL] on each of its 8 channels with modulation possible via FTM2_CH1. See SIM Block Guide for further information.

When FTM1_CH1 is used to modulate an FTM0 channel, then the user must configure FTM1_CH1 to provide a signal that has a higher frequency than the modulated FTM0 channel output. Also it limits the use of the FTM1_CH0 function, as the FTM1_CH1 will be programmed to provide a 50% duty PWM signal and limit the start and modulus values for the free running counter. FTM2 has a similar restriction when FTM2_CH1 is used for modulating an FTM3 channel.

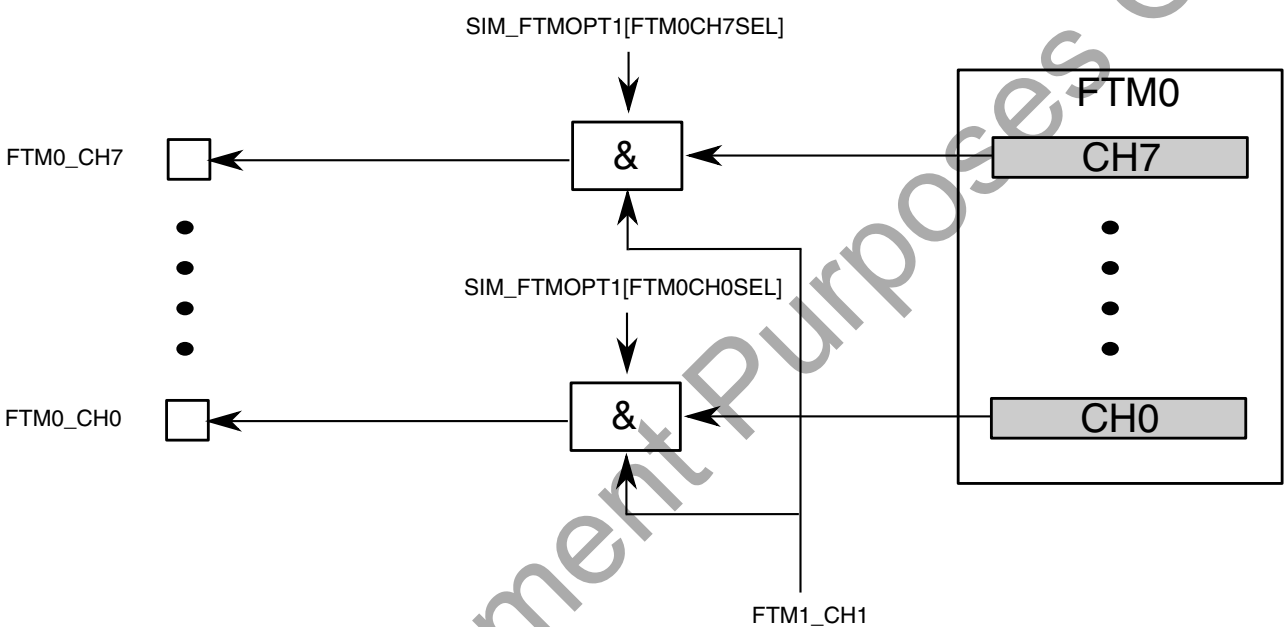


Figure 38-2. FTM Output Modulation

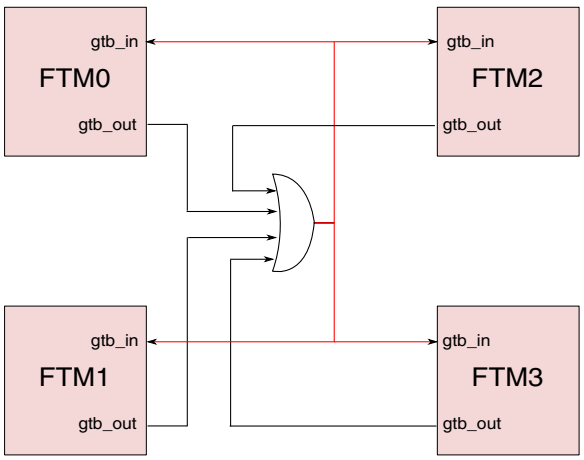
38.1.8 FTM Global Time Base

This chip provides the optional FTM global time base feature, see Global time base (GTB).

FTM supports global timer base through the GTB feature. Any of the FTM module could be used as the GTB_EN source. The global timer base only allows the FTM counters to start their operation synchronously, it does not automatically provide continuous synchronization of FTM counters, meaning that the FTM counters may lose synchronization during misc FTM operation.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required



38.1.9 FTM BDM and debug halt mode

In the FTM chapter, references to the chip being in "BDM" are the same as the chip being in “debug halt mode".

38.2 Introduction

NOTE

The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance. For example, if a module instance supports only six channels, references to channel numbers 6 and 7 do not apply for that instance.

The FlexTimer module (FTM) is a two-to-eight channel timer that supports input capture, output compare, and the generation of PWM signals to control electric motor and power management applications. The FTM time reference is a 16-bit counter that can be used as an unsigned or signed counter.

38.2.1 FlexTimer philosophy

The FlexTimer is built upon a simple timer, the HCS08 Timer PWM Module – TPM, used for many years on Freescale's 8-bit microcontrollers. The FlexTimer extends the functionality to meet the demands of motor control, digital lighting solutions, and power conversion, while providing low cost and backwards compatibility with the TPM module.

For Assessment Purposes Only

Several key enhancements are made:

- Signed up counter
- Deadtime insertion hardware
- Fault control inputs
- Enhanced triggering functionality
- Initialization and polarity control

All of the features common with the TPM have fully backwards compatible register assignments. The FlexTimer can also use code on the same core platform without change to perform the same functions.

Motor control and power conversion features have been added through a dedicated set of registers and defaults turn off all new features. The new features, such as hardware deadtime insertion, polarity, fault control, and output forcing and masking, greatly reduce loading on the execution software and are usually each controlled by a group of registers.

FlexTimer input triggers can be from comparators, ADC, or other submodules to initiate timer functions automatically. These triggers can be linked in a variety of ways during integration of the sub modules so please note the options available for used FlexTimer configuration.

More than one FlexTimers may be synchronized to provide a larger timer with their counters incrementing in unison, assuming the initialization, the input clocks, the initial and final counting values are the same in each FlexTimer.

All main user access registers are buffered to ease the load on the executing software. A number of trigger options exist to determine which registers are updated with this user defined data.

38.2.2 Features

The FTM features include:

- FTM source clock is selectable
  - Source clock can be the system clock, the fixed frequency clock, or an external clock
  - Fixed frequency clock is an additional clock input to allow the selection of an on chip clock source other than the system clock
  - Selecting external clock connects FTM clock to a chip level input pin therefore allowing to synchronize the FTM counter with an off chip clock source

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128
- 16-bit counter
  - It can be a free-running counter or a counter with initial and final value
  - The counting can be up or up-down
- Each channel can be configured for input capture, output compare, or edge-aligned PWM mode
- In Input Capture mode:
  - The capture can occur on rising edges, falling edges or both edges
  - An input filter can be selected for some channels. One unique prescaler is available for all filters
- In Output Compare mode the output signal can be set, cleared, or toggled on match
- All channels can be configured for center-aligned PWM mode
- Each pair of channels can be combined to generate a PWM signal with independent control of both edges of PWM signal
- The FTM channels can operate as pairs with equal outputs, pairs with complementary outputs, or independent channels with independent outputs
- The deadtime insertion is available for each complementary pair
- Generation of match triggers
- Software control of PWM outputs
- Up to 4 fault inputs for global fault control
- The polarity of each channel is configurable
- The generation of an interrupt per channel
- The generation of an interrupt when the counter overflows
- The generation of an interrupt when the fault condition is detected
- The generation of an interrupt when a register reload point occurs
- Synchronized loading of write buffered FTM registers
- Half cycle and Full cycle register reload capacity
- Write protection for critical registers

For Assessment Purposes Only

- Backwards compatible with TPM
- Testing of input capture mode
- Direct access to input pin states
- Dual edge capture for pulse and period width measurement
- Quadrature decoder with prescaled input filters, relative position counting, and interrupt on position count or capture of position count on external event
- The FTM channels can be selected to generate a trigger pulse on channel output instead of a PWM

38.2.3 Modes of operation

When the chip is in an active Debug mode, the FTM temporarily suspends all counting until the chip returns to normal user operating mode. During Stop mode, all FTM input clocks are stopped, so the FTM is effectively disabled until clocks resume. During Wait mode, the FTM continues to operate normally. If the FTM does not need to produce a real time reference or provide the interrupt sources needed to wake the chip from Wait mode, the power can then be saved by disabling FTM functions before entering Wait mode.

38.2.4 Block Diagram

The FTM uses one input/output (I/O) pin per channel, CHn (FTM channel (n)) where n is the channel number (0–7).

NOTE

The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance. For example, if a module instance supports only six channels, references to channel numbers 6 and 7 do not apply for that instance.

The following figure shows the FTM structure. The central component of the FTM is the 16-bit counter with programmable initial and final values and its counting can be up or up-down.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

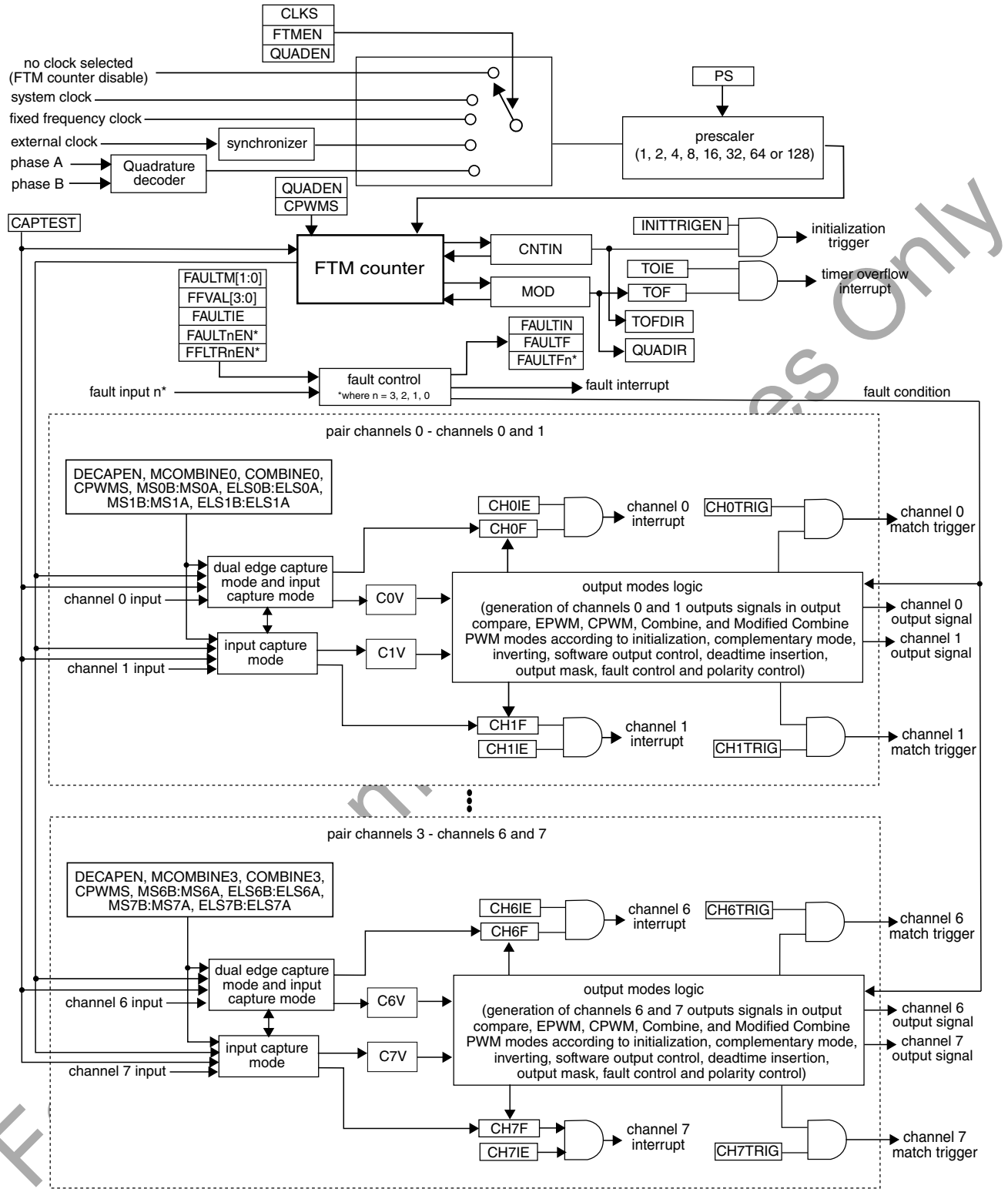


Figure 38-3. FTM Block Diagram

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.3 FTM signal descriptions

Table 38-2 shows the user-accessible signals for the FTM.

Table 38-2. FTM signal descriptions

Signal	Description	I/O	Function
EXTCLK	External clock. FTM external clock can be selected to drive the FTM counter.	I	The external clock input signal is used as the FTM counter clock if selected by CLKS[1:0] bits in the SC register. This clock signal must not exceed 1/4 of system clock frequency. The FTM counter prescaler selection and settings are also used when an external clock is selected.
CHn	FTM channel (n), where n can be 7-0	I/O	Each FTM 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.
FAULTj	Fault input (j), where j can be 3-0	I	The fault input signals are used to control the CHn channel output state. If a fault is detected, the FAULTj signal is asserted and the channel output is put in a safe state. The behavior of the fault logic is defined by the FAULTM[1:0] control bits in the MODE register and FAULTEN bit in the COMBINE register. Note that each FAULTj input may affect all channels selectively since FAULTM[1:0] and FAULTEN control bits are defined for each pair of channels. Because there are several FAULTj inputs, maximum of 4 for the FTM module, each one of these inputs is activated by the FAULTjEN bit in the FLTCTRL register.
PHA	Quadrature decoder phase A input. Input pin associated with quadrature decoder phase A.	I	The quadrature decoder phase A input is used as the Quadrature Decoder mode is selected. The phase A input signal is one of the signals that control the FTM counter increment or decrement in the Quadrature Decoder mode.
PHB	Quadrature decoder phase B input. Input pin associated with quadrature decoder phase B.	I	The quadrature decoder phase B input is used as the Quadrature Decoder mode is selected. The phase B input signal is one of the signals that control the FTM counter increment or decrement in the Quadrature Decoder mode.

For Ass

38.4 Memory map and register definition

38.4.1 Memory map

This section presents a high-level summary of the FTM registers and how they are mapped.

The registers and bits of an unavailable function in the FTM remain in the memory map and in the reset value, but they have no active function.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

NOTE

The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance. For example, if a module instance supports only six channels, references to channel numbers 6 and 7 do not apply for that instance.

Note

Do not write in the region from the CNTIN register through the PWMLOAD register when FTMEN = 0.

38.4.2 Register descriptions

Accesses to reserved addresses result in transfer errors. Registers for absent channels are considered reserved. Double buffered register writes must be done using 32-bit operations.

FTM memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Status And Control (FTM_SC)	32	R/W	0000_0000h	38.4.3/803
4	Counter (FTM_CNT)	32	R/W	0000_0000h	38.4.4/806
8	Modulo (FTM_MOD)	32	R/W	0000_0000h	38.4.5/807
C	Channel (n) Status And Control (FTM_C0SC)	32	R/W	0000_0000h	38.4.6/808
10	Channel (n) Value (FTM_C0V)	32	R/W	0000_0000h	38.4.7/810
14	Channel (n) Status And Control (FTM_C1SC)	32	R/W	0000_0000h	38.4.6/808
18	Channel (n) Value (FTM_C1V)	32	R/W	0000_0000h	38.4.7/810
1C	Channel (n) Status And Control (FTM_C2SC)	32	R/W	0000_0000h	38.4.6/808
20	Channel (n) Value (FTM_C2V)	32	R/W	0000_0000h	38.4.7/810
24	Channel (n) Status And Control (FTM_C3SC)	32	R/W	0000_0000h	38.4.6/808
28	Channel (n) Value (FTM_C3V)	32	R/W	0000_0000h	38.4.7/810
2C	Channel (n) Status And Control (FTM_C4SC)	32	R/W	0000_0000h	38.4.6/808
30	Channel (n) Value (FTM_C4V)	32	R/W	0000_0000h	38.4.7/810
34	Channel (n) Status And Control (FTM_C5SC)	32	R/W	0000_0000h	38.4.6/808
38	Channel (n) Value (FTM_C5V)	32	R/W	0000_0000h	38.4.7/810
3C	Channel (n) Status And Control (FTM_C6SC)	32	R/W	0000_0000h	38.4.6/808
40	Channel (n) Value (FTM_C6V)	32	R/W	0000_0000h	38.4.7/810
44	Channel (n) Status And Control (FTM_C7SC)	32	R/W	0000_0000h	38.4.6/808
48	Channel (n) Value (FTM_C7V)	32	R/W	0000_0000h	38.4.7/810

Table continues on the next page...

Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4C	Counter Initial Value (FTM_CNTIN)	32	R/W	0000_0000h	38.4.8/811
50	Capture And Compare Status (FTM_STATUS)	32	R/W	0000_0000h	38.4.9/811
54	Features Mode Selection (FTM_MODE)	32	R/W	0000_0004h	38.4.10/ 813
58	Synchronization (FTM_SYNC)	32	R/W	0000_0000h	38.4.11/ 815
5C	Initial State For Channels Output (FTM_OUTINIT)	32	R/W	0000_0000h	38.4.12/ 817
60	Output Mask (FTM_OUTMASK)	32	R/W	0000_0000h	38.4.13/ 819
64	Function For Linked Channels (FTM_COMBINE)	32	R/W	0000_0000h	38.4.14/ 821
68	Deadtime Insertion Control (FTM_DEADTIME)	32	R/W	0000_0000h	38.4.15/ 825
6C	FTM External Trigger (FTM_EXTTRIG)	32	R/W	0000_0000h	38.4.16/ 826
70	Channels Polarity (FTM_POL)	32	R/W	0000_0000h	38.4.17/ 829
74	Fault Mode Status (FTM_FMS)	32	R/W	0000_0000h	38.4.18/ 831
78	Input Capture Filter Control (FTM_FILTER)	32	R/W	0000_0000h	38.4.19/ 833
7C	Fault Control (FTM_FLTCTRL)	32	R/W	0000_0000h	38.4.20/ 834
80	Quadrature Decoder Control And Status (FTM_QDCTRL)	32	R/W	0000_0000h	38.4.21/ 837
84	Configuration (FTM_CONF)	32	R/W	0000_0000h	38.4.22/ 839
88	FTM Fault Input Polarity (FTM_FLTPOL)	32	R/W	0000_0000h	38.4.23/ 840
8C	Synchronization Configuration (FTM_SYNCONF)	32	R/W	0000_0000h	38.4.24/ 841
90	FTM Inverting Control (FTM_INVCTRL)	32	R/W	0000_0000h	38.4.25/ 843
94	FTM Software Output Control (FTM_SWOCTRL)	32	R/W	0000_0000h	38.4.26/ 844
98	FTM PWM Load (FTM_PWMLOAD)	32	R/W	0000_0000h	38.4.27/ 847
9C	Half Cycle Register (FTM_HCR)	32	R/W	0000_0000h	38.4.28/ 849

Freescale Confidential Proprietary - Non-Disclosure Agreement required

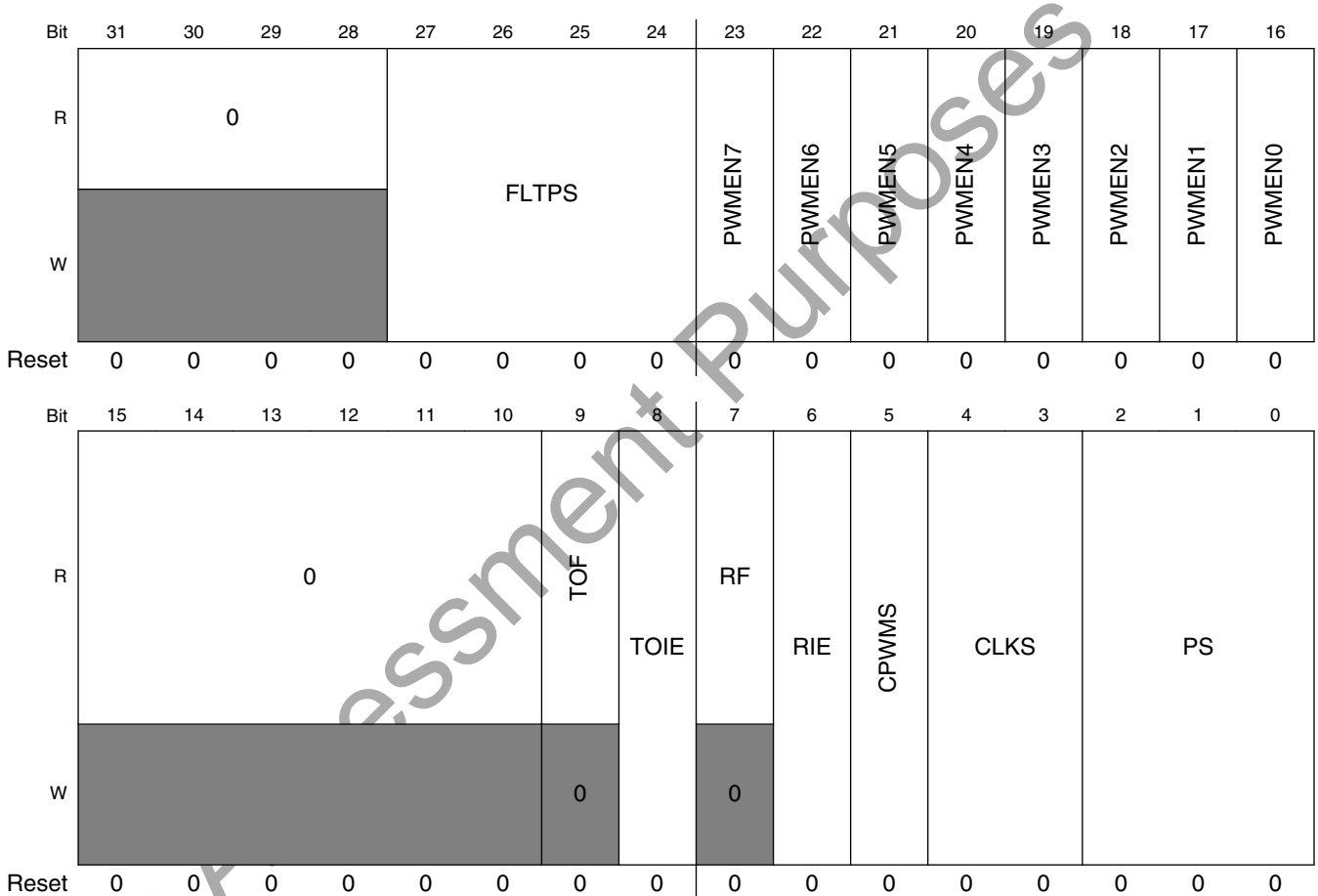
38.4.3 Status And Control (FTM_SC)

SC contains the overflow status flag and control bits used to configure the interrupt enable, FTM configuration, clock source, filter prescaler, and prescaler factor.

This register also contains the output enable control bits and the reload opportunity flag control.

These controls relate to all channels within this module.

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



FTM_SC 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 FLTPS	Filter Prescaler Selects one of 16 division factors for the clock used at input filters. The new prescaler factor has effect on the next system clock cycle after the new value is updated into the register bits. The filter prescaler affects the channel input filters, quadrature input filters and fault control input filters.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Assessment Purposes Only

FTM_SC field descriptions (continued)

Field	Description
	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
23 PWMEN7	Channel 7 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0 Channel output port is disabled 1 Channel output port is enabled
22 PWMEN6	Channel 6 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0 Channel output port is disabled 1 Channel output port is enabled
21 PWMEN5	Channel 5 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0 Channel output port is disabled 1 Channel output port is enabled
20 PWMEN4	Channel 4 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0 Channel output port is disabled 1 Channel output port is enabled
19 PWMEN3	Channel 3 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0 Channel output port is disabled 1 Channel output port is enabled

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_SC field descriptions (continued)

Field	Description
18 PWMEN2	Channel 2 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0 Channel output port is disabled 1 Channel output port is enabled
17 PWMEN1	Channel 1 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0 Channel output port is disabled 1 Channel output port is enabled
16 PWMEN0	Channel 0 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0 Channel output port is disabled 1 Channel output port is enabled
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 TOF	Timer Overflow Flag Set by hardware when the FTM counter passes the value in the MOD register. The TOF bit is cleared by reading the SC register while TOF is set and then writing a 0 to TOF bit. Writing a 1 to TOF has no effect. If another FTM overflow occurs between the read and write operations, the write operation has no effect; therefore, TOF remains set indicating an overflow has occurred. In this case, a TOF interrupt request is not lost due to the clearing sequence for a previous TOF. 0 FTM counter has not overflowed. 1 FTM counter has overflowed.
8 TOIE	Timer Overflow Interrupt Enable Enables FTM overflow interrupts. 0 Disable TOF interrupts. Use software polling. 1 Enable TOF interrupts. An interrupt is generated when TOF equals one.
7 RF	Reload Flag Set by hardware when FTM counter matches the value of a reload point configured by FTMxPWMLOAD register. The RF bit is cleared by reading the SC register while RF is set and then writing a 0 to RF bit. Writing 1 to RF has no effect. If another reload point is reached between the read and write operations, the write operation has no effect; therefore, RF remains set. 0 FTM counter did not reach a reload point. 1 FTM counter reached a reload point.
6 RIE	Reload Interrupt Enable Enables the reload oportunity interrupt.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_SC field descriptions (continued)

Field	Description
	0 Reload interrupt is disabled. 1 Reload interrupt is enabled.
5 CPWMS	Center-Aligned PWM Select Selects CPWM mode. This mode configures the FTM to operate in Up-Down Counting mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 FTM counter operates in Up Counting mode. 1 FTM counter operates in Up-Down Counting mode.
4–3 CLKS	Clock Source Selection Selects one of the three FTM counter clock sources. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00 No clock selected. This in effect disables the FTM counter. 01 System clock 10 Fixed frequency clock 11 External clock
PS	Prescale Factor Selection Selects one of 8 division factors for the clock source selected by CLKS. The new prescaler factor affects the clock source on the next system clock cycle after the new value is updated into the register bits. This field is write protected. It can be written only when MODE[WPDIS] = 1. 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

For Asse

38.4.4 Counter (FTM_CNT)

The CNT register contains the FTM counter value.

Reset clears the CNT register. Writing any value to COUNT updates the counter with its initial value, CNTIN.

Address: 0h base + 4h offset = 4h

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_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

38.4.5 Modulo (FTM_MOD)

The Modulo register contains the modulo value for the FTM counter. After the FTM counter reaches the modulo value, the overflow flag (TOF) becomes set at the next clock cycle, and the next value of FTM counter depends on the selected counting method; see Counter.

Writes to the MOD register are done on its write buffer. The MOD register is updated with its write buffer value according to Registers updated from write buffers. If FTMEN = 0, a write to SC register resets manually this write coherency mechanism.

Initialize the FTM counter, by writing to CNT, before writing to the MOD register to avoid confusion about when the first counter overflow will occur.

Address: 0h base + 8h offset = 8h

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

FTM_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

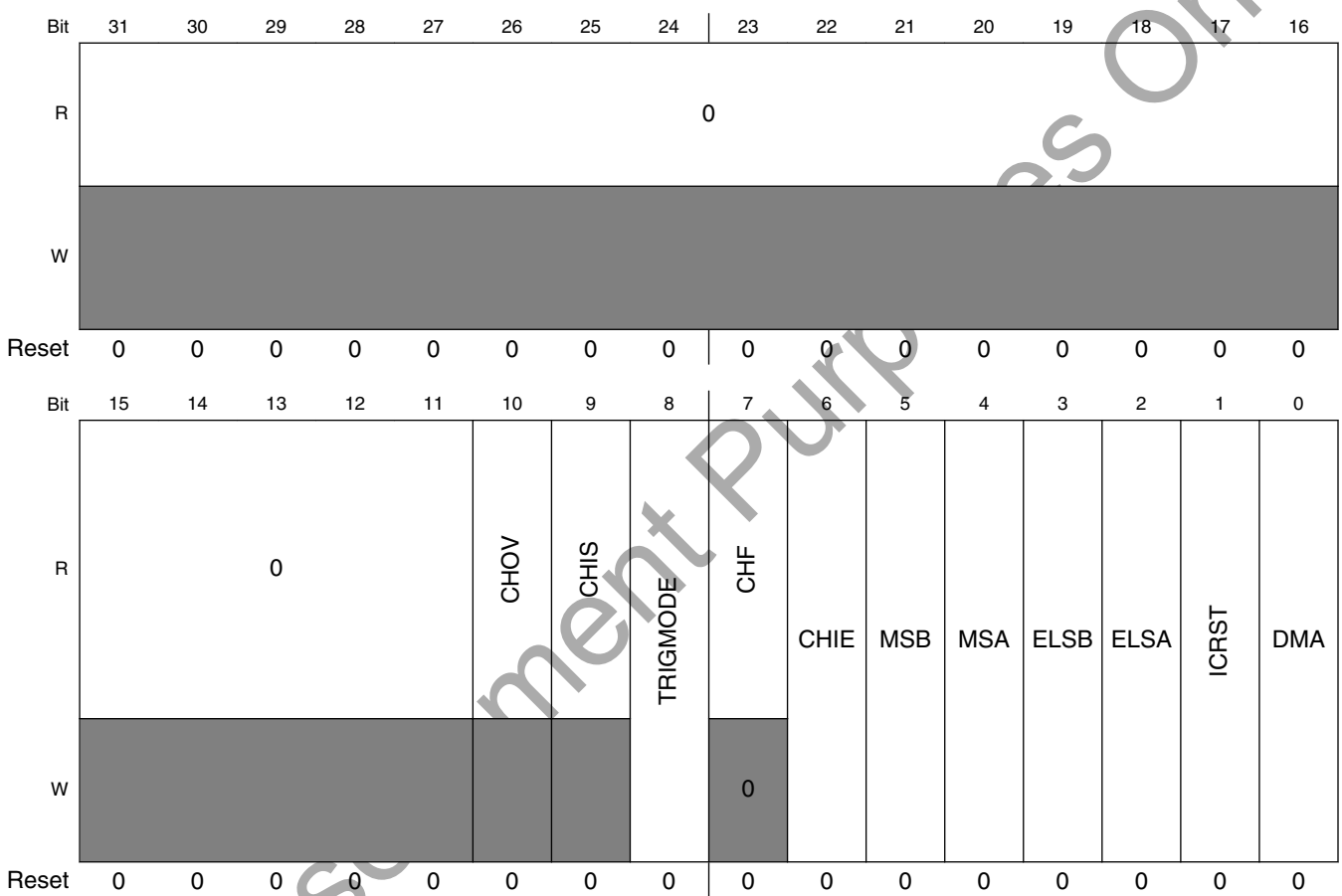
For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.4.6 Channel (n) Status And Control (FTM_C*n*SC)

CnSC contains channel (n) status bits and control bits that select the channel (n) mode and its functionality.

Address: 0h base + Ch offset + (8d × i), where i=0d to 7d



Document Purposes Only

FTM_C*n*SC field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 CHOV	Channel (n) Output Value The CHOV bit has the final value of the channel (n) output. **NOTE:** The CHOV bit should be ignored when the channel (n) is not in an output mode. 0 The channel (n) output is zero. 1 The channel (n) output is one.
9 CHIS	Channel (n) Input State

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_C*n*SC field descriptions (continued)

Field	Description
	The CHIS bit has the value of the channel (n) input after the double-sampling or the filtering (if the channel (n) filter is enabled) both them are inside the FTM. **NOTE:** The CHIS bit should be ignored when the channel (n) is not in an input mode. **NOTE:** When the pair channels is on dual edge mode, the channel (n+1) CHIS bit is the channel (n+1) input value and not the channel (n) input value (this signal is the input signal used by the dual edge mode). 0 The channel (n) input is zero. 1 The channel (n) input is one.
8 TRIGMODE	Trigger mode control This bit controls the trigger generation on FTM channel outputs. This mode is allowed only if when FTM channel is configured to EPWM (up counting) or CPWM (up-down counting) modes. If a match in the channel occurs, a trigger pulse with one FTM clock cycle width will be generated in the channel output. See Channel trigger output for more details about trigger mode feature. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channel outputs will generate the normal PWM outputs without generating a pulse. 1 If a match in the channel occurs, a trigger generation on channel output will happen. The trigger pulse width has one FTM clock cycle.
7 CHF	Channel (n) Flag Set by hardware when an event occurs on the channel (n). CHF is cleared by reading the CnSC register while CHF is set and then writing a 0 to the CHF bit. Writing a 1 to CHF has no effect. If another event occurs between the read and write operations, the write operation has no effect; therefore, CHF remains set indicating an event has occurred. In this case a CHF interrupt request is not lost due to the clearing sequence for a previous CHF. 0 No channel (n) event has occurred. 1 A channel (n) event has occurred.
6 CHIE	Channel (n) Interrupt Enable Enables channel (n) interrupt. 0 Disable channel (n) interrupt. Use software polling. 1 Enable channel (n) interrupt.
5 MSB	Channel (n) Mode Select Used on the selection of the channel (n) mode. See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
4 MSA	Channel (n) Mode Select Used on the selection of the channel (n) mode. See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
3 ELSB	Channel (n) Edge or Level Select Used on the selection of the channel (n) mode. See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
2 ELSA	Channel (n) Edge or Level Select Used on the selection of the channel (n) mode. See Channel Modes.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_C*n*SC field descriptions (continued)

Field	Description
	This field is write protected. It can be written only when MODE[WPDIS] = 1.
1 ICRST	FTM counter reset by the selected input capture event. FTM counter reset is driven by the selected event of the channel (n) in the Input Capture mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 FTM counter is not reset when the selected channel (n) input event is detected. 1 FTM counter is reset when the selected channel (n) input event is detected.
0 DMA	DMA Enable Enables DMA transfers for the channel. 0 Disable DMA transfers. 1 Enable DMA transfers.

38.4.7 Channel (n) Value (FTM_C*n*V)

These registers contain the captured FTM counter value for the input modes or the match value for the output modes.

In Input Capture, Capture Test, and Dual Edge Capture modes, any write to a CnV register is ignored.

In output modes, writes to the CnV register are done on its write buffer. The CnV register is updated with its write buffer value according to Registers updated from write buffers. If FTMEN = 0, a write to CnSC register resets manually this write coherency mechanism.

Address: 0h base + 10h offset + (8d × i), where i=0d to 7d

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

FTM_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 FTM counter value of the input modes or the match value for the output modes

Assessment Purpose

38.4.8 Counter Initial Value (FTM_CNTIN)

The Counter Initial Value register contains the initial value for the FTM counter.

Writing to the CNTIN register latches the value into a buffer. The CNTIN register is updated with the value of its write buffer according to Registers updated from write buffers.

When the FTM clock is initially selected, by writing a non-zero value to the CLKS bits, the FTM counter starts with the value 0x0000. To avoid this behavior, before the first write to select the FTM clock, write the new value to the the CNTIN register and then initialize the FTM counter by writing any value to the CNT register.

Address: 0h base + 4Ch offset = 4Ch

Bit	31	30	29	28	27	26	25	24	23	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																INIT															
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

FTM_CNTIN field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
INIT	Initial Value Of The FTM Counter

38.4.9 Capture And Compare Status (FTM_STATUS)

The STATUS register contains a copy of the status flag CHnF bit in CnSC for each FTM channel 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 reading STATUS followed by writing 0x00 to STATUS.

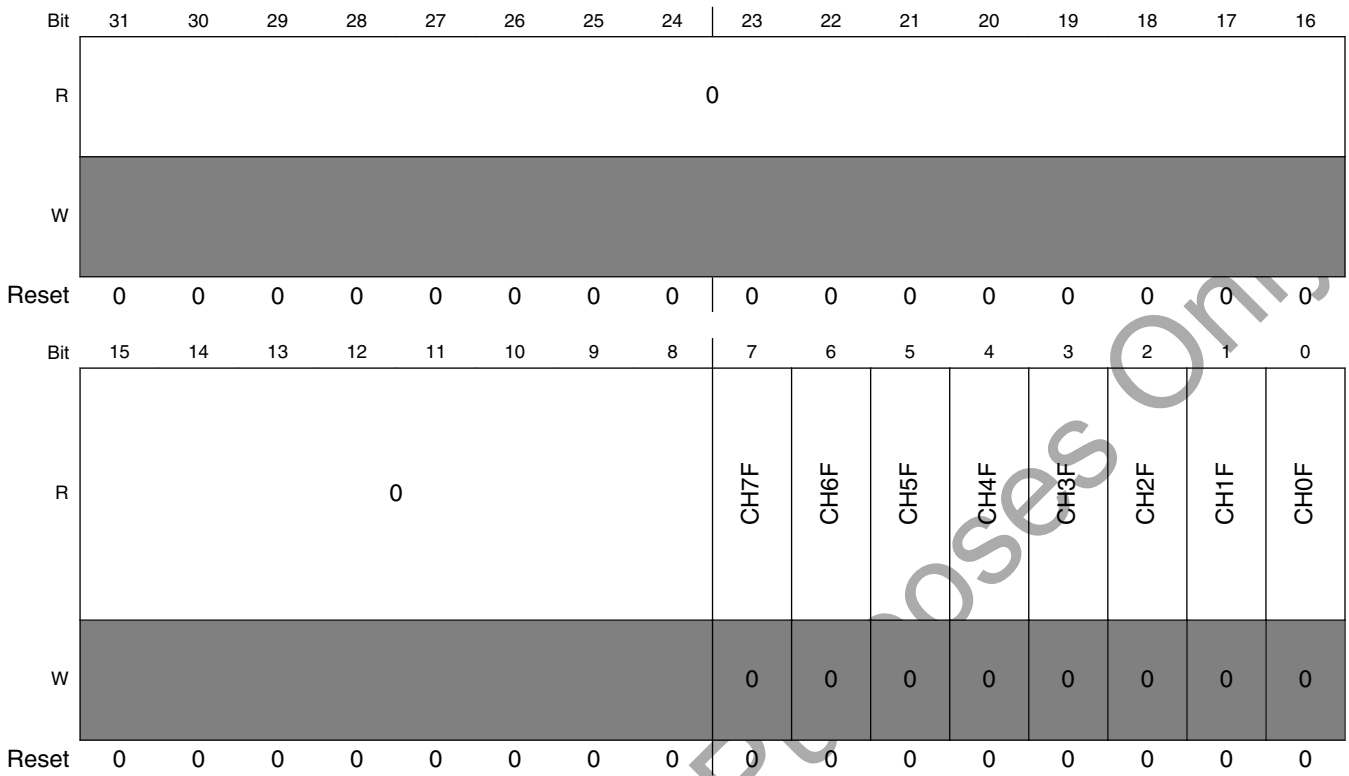
Hardware sets the individual channel flags when an event occurs on the channel. CHnF is cleared by reading STATUS while CHnF is set and then writing a 0 to the CHnF bit. Writing a 1 to CHnF has no effect.

If another event occurs between the read and write operations, the write operation has no effect; therefore, CHnF remains set indicating an event has occurred. In this case, a CHnF interrupt request is not lost due to the clearing sequence for a previous CHnF.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 50h offset = 50h



Purposes Only

FTM_STATUS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7F	Channel 7 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
6 CH6F	Channel 6 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
5 CH5F	Channel 5 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
4 CH4F	Channel 4 Flag See the register description.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_STATUS field descriptions (continued)

Field	Description
	0 No channel event has occurred. 1 A channel event has occurred.
3 CH3F	Channel 3 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
2 CH2F	Channel 2 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
1 CH1F	Channel 1 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
0 CH0F	Channel 0 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.

For Assessment

38.4.10 Features Mode Selection (FTM_MODE)

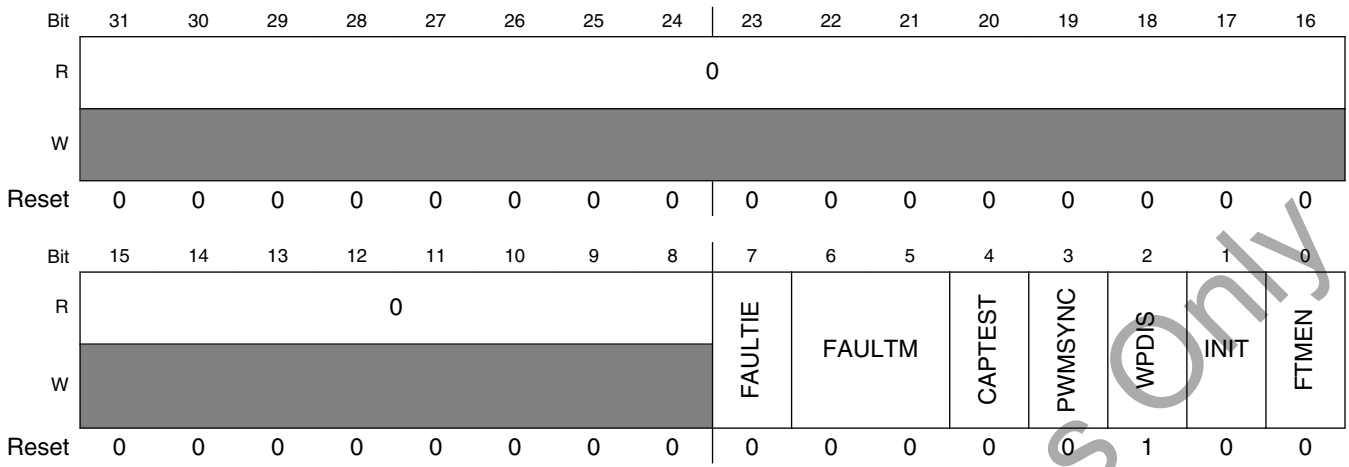
This register contains the global enable bit for FTM-specific features and the control bits used to configure:

- Fault control mode and interrupt
- Capture Test mode
- PWM synchronization
- Write protection
- Channel output initialization

These controls relate to all channels within this module.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 54h offset = 54h



FTM_MODE field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 FAULTIE	Fault Interrupt Enable Enables the generation of an interrupt when a fault is detected by FTM and the FTM fault control is enabled. 0 Fault control interrupt is disabled. 1 Fault control interrupt is enabled.
6–5 FAULTM	Fault Control Mode Defines the FTM fault control mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00 Fault control is disabled for all channels. 01 Fault control is enabled for even channels only (channels 0, 2, 4, and 6), and the selected mode is the manual fault clearing. 10 Fault control is enabled for all channels, and the selected mode is the manual fault clearing. 11 Fault control is enabled for all channels, and the selected mode is the automatic fault clearing.
4 CAPTEST	Capture Test Mode Enable Enables the capture test mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Capture test mode is disabled. 1 Capture test mode is enabled.
3 PWMSYNC	PWM Synchronization Mode Selects which triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. See PWM synchronization. The PWMSYNC bit configures the synchronization when SYNCMODE is 0. 0 No restrictions. Software and hardware triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. 1 Software trigger can only be used by MOD and CnV synchronization, and hardware triggers can only be used by OUTMASK and FTM counter synchronization.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_MODE field descriptions (continued)

Field	Description
2 WPDIS	Write Protection Disable When write protection is enabled (WPDIS = 0), write protected bits cannot be written. When write protection is disabled (WPDIS = 1), write protected bits can be written. The WPDIS bit is the negation of the WPEN bit. WPDIS is cleared when 1 is written to WPEN. WPDIS is set when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPDIS has no effect. 0 Write protection is enabled. 1 Write protection is disabled.
1 INIT	Initialize The Channels Output When a 1 is written to INIT bit the channels output is initialized according to the state of their corresponding bit in the OUTINIT register. Writing a 0 to INIT bit has no effect. The INIT bit is always read as 0.
0 FTMEN	FTM Enable This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 TPM compatibility. Free running counter and synchronization compatible with TPM. 1 Free running counter and synchronization are different from TPM behavior.

38.4.11 Synchronization (FTM_SYNC)

This register configures the PWM synchronization.

A synchronization event can perform the synchronized update of MOD, CV, and OUTMASK registers with the value of their write buffer and the FTM counter initialization.

NOTE

The software trigger, SWSYNC bit, and hardware triggers TRIG0, TRIG1, and TRIG2 bits have a potential conflict if used together when SYNCMODE = 0. Use only hardware or software triggers but not both at the same time, otherwise unpredictable behavior is likely to happen.

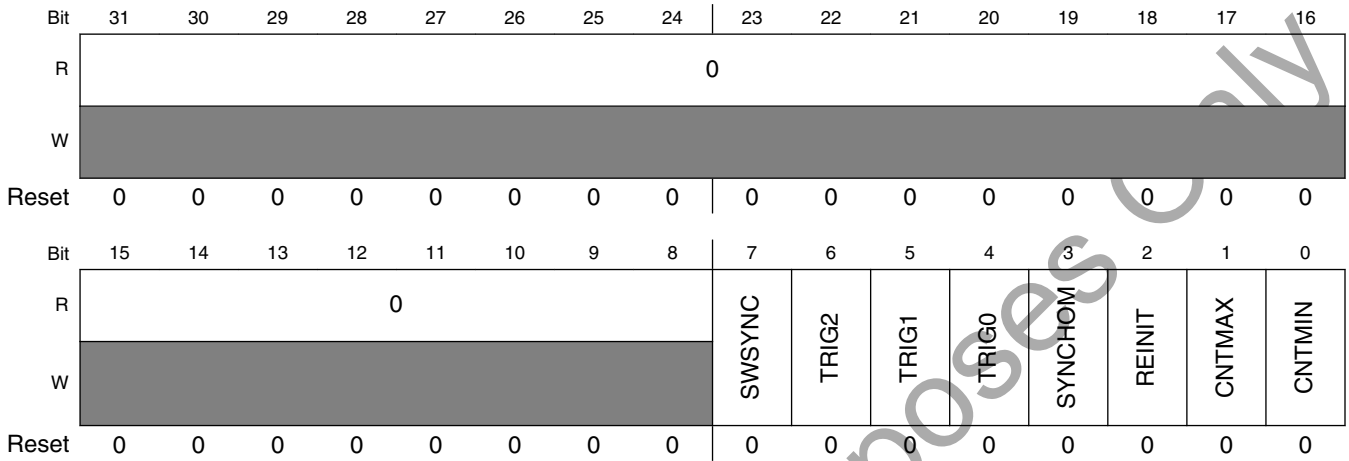
The selection of the loading point, CNTMAX and CNTMIN bits, is intended to provide the update of MOD, CNTIN, and CnV registers across all enabled channels simultaneously. The use of the loading point selection together with SYNCMODE = 0 and hardware trigger selection, TRIG0, TRIG1, or TRIG2 bits, is likely to result in unpredictable behavior.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The synchronization event selection also depends on the PWMSYNC (MODE register) and SYNCMODE (SYNCONF register) bits. See PWM synchronization.

Address: 0h base + 58h offset = 58h



Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_SYNC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 SWSYNC	PWM Synchronization Software Trigger Selects the software trigger as the PWM synchronization trigger. The software trigger happens when a 1 is written to SWSYNC bit. 0 Software trigger is not selected. 1 Software trigger is selected.
6 TRIG2	PWM Synchronization Hardware Trigger 2 Enables hardware trigger 2 to the PWM synchronization. Hardware trigger 2 happens when a rising edge is detected at the trigger 2 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
5 TRIG1	PWM Synchronization Hardware Trigger 1 Enables hardware trigger 1 to the PWM synchronization. Hardware trigger 1 happens when a rising edge is detected at the trigger 1 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
4 TRIG0	PWM Synchronization Hardware Trigger 0 Enables hardware trigger 0 to the PWM synchronization. Hardware trigger 0 occurs when a rising edge is detected at the trigger 0 input signal.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_SYNC field descriptions (continued)

Field	Description
	0 Trigger is disabled. 1 Trigger is enabled.
3 SYNCHOM	Output Mask Synchronization Selects when the OUTMASK register is updated with the value of its buffer. 0 OUTMASK register is updated with the value of its buffer in all rising edges of the system clock. 1 OUTMASK register is updated with the value of its buffer only by the PWM synchronization.
2 REINIT	FTM Counter Reinitialization By Synchronization (FTM counter synchronization) Determines if the FTM counter is reinitialized when the selected trigger for the synchronization is detected. The REINIT bit configures the synchronization when SYNCMODE is zero. 0 FTM counter continues to count normally. 1 FTM counter is updated with its initial value when the selected trigger is detected.
1 CNTMAX	Maximum Loading Point Enable Selects the maximum loading point to PWM synchronization. See Boundary cycle and loading points. If CNTMAX is 1, the selected loading point is when the FTM counter reaches its maximum value (MOD register). 0 The maximum loading point is disabled. 1 The maximum loading point is enabled.
0 CNTMIN	Minimum Loading Point Enable Selects the minimum loading point to PWM synchronization. See Boundary cycle and loading points. If CNTMIN is one, the selected loading point is when the FTM counter reaches its minimum value (CNTIN register). 0 The minimum loading point is disabled. 1 The minimum loading point is enabled.

38.4.12 Initial State For Channels Output (FTM_OUTINIT)

Address: 0h base + 5Ch offset = 5Ch

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								CH7OI	CH6OI	CH5OI	CH4OI	CH3OI	CH2OI	CH1OI	CH0OI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

For Assess

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_OUTINIT field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7OI	Channel 7 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
6 CH6OI	Channel 6 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
5 CH5OI	Channel 5 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
4 CH4OI	Channel 4 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
3 CH3OI	Channel 3 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
2 CH2OI	Channel 2 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
1 CH1OI	Channel 1 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
0 CH0OI	Channel 0 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.4.13 Output Mask (FTM_OUTMASK)

This register provides a mask for each FTM channel. The mask of a channel determines if its output responds, that is, it is masked or not, when a match occurs. This feature is used for BLDC control where the PWM signal is presented to an electric motor at specific times to provide electronic commutation.

Any write to the OUTMASK register, stores the value in its write buffer. The register is updated with the value of its write buffer according to PWM synchronization.

Output Mask bits must not be set for trigger mode.

Address: 0h base + 60h offset = 60h

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								CH7OM	CH6OM	CH5OM	CH4OM	CH3OM	CH2OM	CH1OM	CH0OM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTM_OUTMASK field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7OM	Channel 7 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
6 CH6OM	Channel 6 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
5 CH5OM	Channel 5 Output Mask Defines if the channel output is masked or unmasked.

Table continues on the next page...

Content Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_OUTMASK field descriptions (continued)

Field	Description
	0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
4 CH4OM	Channel 4 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
3 CH3OM	Channel 3 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
2 CH2OM	Channel 2 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
1 CH1OM	Channel 1 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
0 CH0OM	Channel 0 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.

For Assess

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.4.14 Function For Linked Channels (FTM_COMBINE)

This register contains the configuration bits for each pair of channels.

Address: 0h base + 64h offset = 64h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MCOMBINE3	FAULTEN3	SYNCEN3	DTEN3	DECAP3	DECAPEN3	COMP3	COMBINE3	MCOMBINE2	FAULTEN2	SYNCEN2	DTEN2	DECAP2	DECAPEN2	COMP2	COMBINE2
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	MCOMBINE1	FAULTEN1	SYNCEN1	DTEN1	DECAP1	DECAPEN1	COMP1	COMBINE1	MCOMBINE0	FAULTEN0	SYNCEN0	DTEN0	DECAP0	DECAPEN0	COMP0	COMBINE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTM_COMBINE field descriptions

Field	Description
31 MCOMBINE3	Modified Combine Mode For n = 6 Used on the selection of the modified combine mode for channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
30 FAULTEN3	Fault Control Enable For n = 6 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
29 SYNCEN3	Synchronization Enable For n = 6 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
28 DTEN3	Deadtime Enable For n = 6 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

Purposes Only

FTM_COMBINE field descriptions (continued)

Field	Description
	0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
27 DECAP3	Dual Edge Capture Mode Captures For n = 6 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
26 DECAPEN3	Dual Edge Capture Mode Enable For n = 6 Enables the Dual Edge Capture mode in the channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
25 COMP3	Complement Of Channel (n) for n = 6 In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
24 COMBINE3	Combine Channels For n = 6 Used on the selection of the combine mode for channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
23 MCOMBINE2	Modified Combine Mode For n = 4 Used on the selection of the modified combine mode for channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
22 FAULTEN2	Fault Control Enable For n = 4 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
21 SYNCEN2	Synchronization Enable For n = 4 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
20 DTEN2	Deadtime Enable For n = 4 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_COMBINE field descriptions (continued)

Field	Description
	0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
19 DECAP2	Dual Edge Capture Mode Captures For n = 4 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
18 DECAPEN2	Dual Edge Capture Mode Enable For n = 4 Enables the Dual Edge Capture mode in the channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
17 COMP2	Complement Of Channel (n) For n = 4 In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
16 COMBINE2	Combine Channels For n = 4 Used on the selection of the combine mode for channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
15 MCOMBINE1	Modified Combine Mode For n = 2 Used on the selection of the modified combine mode for channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
14 FAULTEN1	Fault Control Enable For n = 2 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
13 SYNCEN1	Synchronization Enable For n = 2 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
12 DTEN1	Deadtime Enable For n = 2 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_COMBINE field descriptions (continued)

Field	Description
	0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
11 DECAP1	Dual Edge Capture Mode Captures For n = 2 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if Dual Edge Capture – One-Shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
10 DECAPEN1	Dual Edge Capture Mode Enable For n = 2 Enables the Dual Edge Capture mode in the channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
9 COMP1	Complement Of Channel (n) For n = 2 In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
8 COMBINE1	Combine Channels For n = 2 Used on the selection of the combine mode for channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
7 MCOMBINE0	Modified Combine Mode For n = 0 Used on the selection of the modified combine mode for channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
6 FAULTEN0	Fault Control Enable For n = 0 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
5 SYNCEN0	Synchronization Enable For n = 0 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
4 DTEN0	Deadtime Enable For n = 0 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

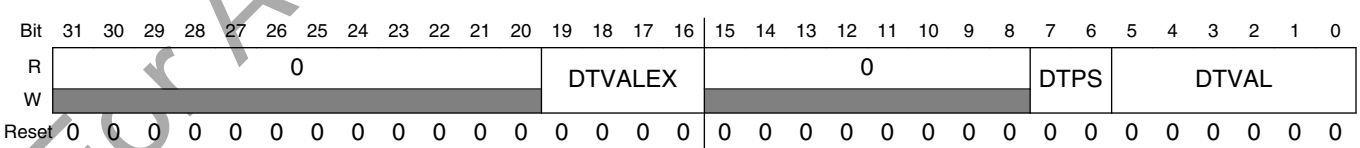
FTM_COMBINE field descriptions (continued)

Field	Description
	0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
3 DECAP0	Dual Edge Capture Mode Captures For n = 0 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
2 DECAPEN0	Dual Edge Capture Mode Enable For n = 0 Enables the Dual Edge Capture mode in the channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.
1 COMP0	Complement Of Channel (n) For n = 0 In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
0 COMBINE0	Combine Channels For n = 0 Used on the selection of the combine mode for channels (n) and (n+1). See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1.

38.4.15 Deadtime Insertion Control (FTM_DEADTIME)

This register selects the deadtime prescaler factor and deadtime value. All FTM channels use this clock prescaler and this deadtime value for the deadtime insertion.

Address: 0h base + 68h offset = 68h



FTM_DEADTIME field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_DEADTIME field descriptions (continued)

Field	Description
19–16 DTVALEX	Extended Deadtime Value This field is a bit extension of the DTVAL field. It defines the 4 most significant bits of the deadtime value. The maximum deadtime value is extended to 1023 using the concatenation {DTVALEX, DTVAL}. Deadtime insert value = (DTPS × {DTVALEX, DTVAL}). This field is write protected. It can be written only when MODE[WPDIS] = 1. Note: If full compatibility is needed with previous software versions, write 0 to DTVALEX bits.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 DTPS	Deadtime Prescaler Value Selects the division factor of the system clock. This prescaled clock is used by the deadtime counter. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0x Divide the system clock by 1. 10 Divide the system clock by 4. 11 Divide the system clock by 16.
DTVAL	Deadtime Value Selects the deadtime insertion value for the deadtime counter. The deadtime counter is clocked by a scaled version of the system clock. See the description of DTPS. This field is write protected. It can be written only when MODE[WPDIS] = 1.

38.4.16 FTM External Trigger (FTM_EXTTRIG)

This register:

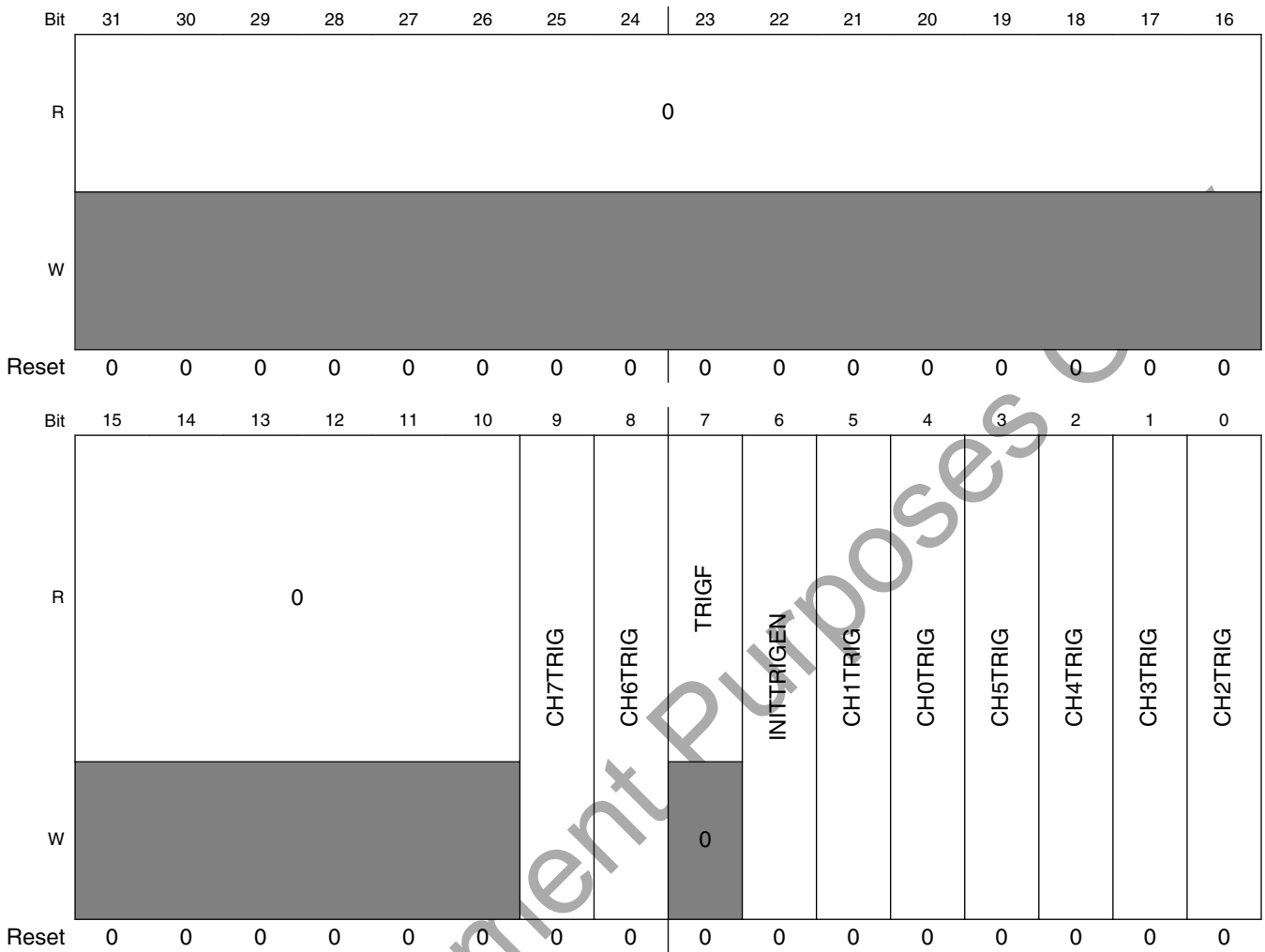
- Indicates when a channel trigger was generated
- Enables the generation of a trigger when the FTM counter is equal to its initial value
- Selects which channels are used in the generation of the channel triggers

Several channels can be selected to generate multiple triggers in one PWM period. See External Trigger and Initialization trigger

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 6Ch offset = 6Ch



FTM_EXTTRIG field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 CH7TRIG	Channel 7 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
8 CH6TRIG	Channel 6 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
7 TRIGF	Channel Trigger Flag

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_EXTTRIG field descriptions (continued)

Field	Description
	Set by hardware when a channel trigger is generated. Clear TRIGF by reading EXTTRIG while TRIGF is set and then writing a 0 to TRIGF. Writing a 1 to TRIGF has no effect. If another channel trigger is generated before the clearing sequence is completed, the sequence is reset so TRIGF remains set after the clear sequence is completed for the earlier TRIGF. 0 No channel trigger was generated. 1 A channel trigger was generated.
6 INITTRIGEN	Initialization Trigger Enable Enables the generation of the trigger when the FTM counter is equal to the CNTIN register. 0 The generation of initialization trigger is disabled. 1 The generation of initialization trigger is enabled.
5 CH1TRIG	Channel 1 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
4 CH0TRIG	Channel 0 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
3 CH5TRIG	Channel 5 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
2 CH4TRIG	Channel 4 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
1 CH3TRIG	Channel 3 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
0 CH2TRIG	Channel 2 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.4.17 Channels Polarity (FTM_POL)

This register defines the output polarity of the FTM channels.

NOTE

The safe value that is driven in a channel output when the fault control is enabled and a fault condition is detected is the inactive state of the channel. That is, the safe value of a channel is the value of its POL bit.

Address: 0h base + 70h offset = 70h

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								POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTM_POL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 POL7	Channel 7 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
6 POL6	Channel 6 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
5 POL5	Channel 5 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
4 POL4	Channel 4 Polarity

Table continues on the next page...

Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_POL field descriptions (continued)

Field	Description
	Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
3 POL3	Channel 3 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
2 POL2	Channel 2 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
1 POL1	Channel 1 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
0 POL0	Channel 0 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.

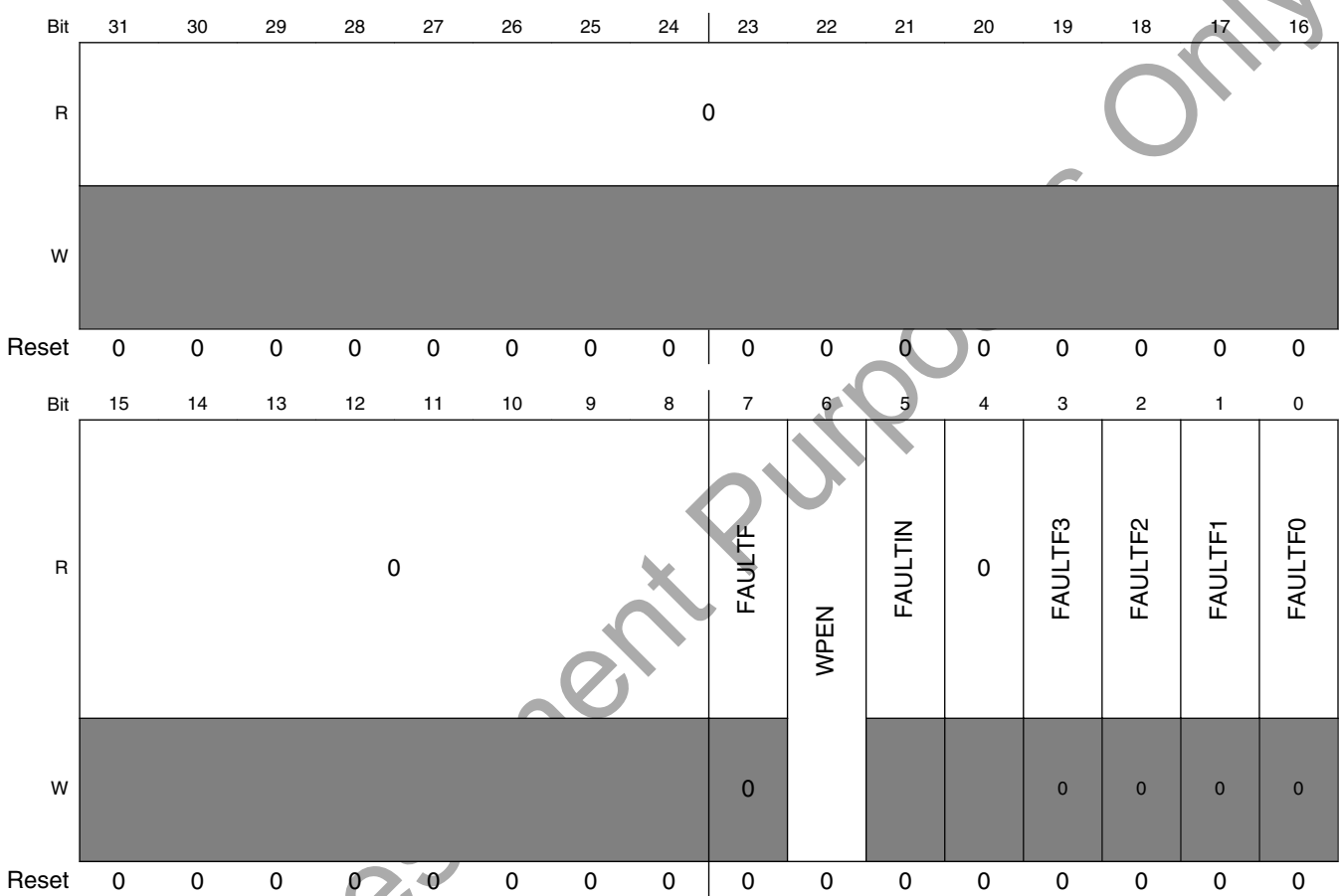
For Asse

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.4.18 Fault Mode Status (FTM_FMS)

This register contains the fault detection flags, write protection enable bit, and the logic OR of the enabled fault inputs.

Address: 0h base + 74h offset = 74h



Freescale Confidential Proprietary - Non-Disclosure Agreement required Only

FTM_FMS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 FAULTF	Fault Detection Flag Represents the logic OR of the individual FAULTFj bits where j = 3, 2, 1, 0. Clear FAULTF by reading the FMS register while FAULTF is set and then writing a 0 to FAULTF while there is no existing fault condition at the enabled fault inputs. Writing a 1 to FAULTF has no effect. If another fault condition is detected in an enabled fault input before the clearing sequence is completed, the sequence is reset so FAULTF remains set after the clearing sequence is completed for the earlier fault condition. FAULTF is also cleared when FAULTFj bits are cleared individually. 0 No fault condition was detected. 1 A fault condition was detected.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_FMS field descriptions (continued)

Field	Description
6 WPEN	Write Protection Enable The WPEN bit is the negation of the WPDIS bit. WPEN is set when 1 is written to it. WPEN is cleared when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPEN has no effect. 0 Write protection is disabled. Write protected bits can be written. 1 Write protection is enabled. Write protected bits cannot be written.
5 FAULTIN	Fault Inputs Represents the logic OR of the enabled fault inputs after their filter (if their filter is enabled) when fault control is enabled. 0 The logic OR of the enabled fault inputs is 0. 1 The logic OR of the enabled fault inputs is 1.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 FAULTF3	Fault Detection Flag 3 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF3 by reading the FMS register while FAULTF3 is set and then writing a 0 to FAULTF3 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF3 has no effect. FAULTF3 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF3 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
2 FAULTF2	Fault Detection Flag 2 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF2 by reading the FMS register while FAULTF2 is set and then writing a 0 to FAULTF2 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF2 has no effect. FAULTF2 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF2 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
1 FAULTF1	Fault Detection Flag 1 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF1 by reading the FMS register while FAULTF1 is set and then writing a 0 to FAULTF1 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF1 has no effect. FAULTF1 bit is also cleared when FAULTF bit is cleared.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_FMS field descriptions (continued)

Field	Description
	If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF1 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
0 FAULTF0	Fault Detection Flag 0 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF0 by reading the FMS register while FAULTF0 is set and then writing a 0 to FAULTF0 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF0 has no effect. FAULTF0 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF0 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.

38.4.19 Input Capture Filter Control (FTM_FILTER)

This register selects the filter value for the inputs of channels.

Channels 4, 5, 6 and 7 do not have an input filter.

NOTE

Writing to the FILTER register has immediate effect and must be done only when the channels 0, 1, 2, and 3 are not in input modes. Failure to do this could result in a missing valid signal.

Address: 0h base + 78h offset = 78h

Bit	31	30	29	28	27	26	25	24	23	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

FTM_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 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

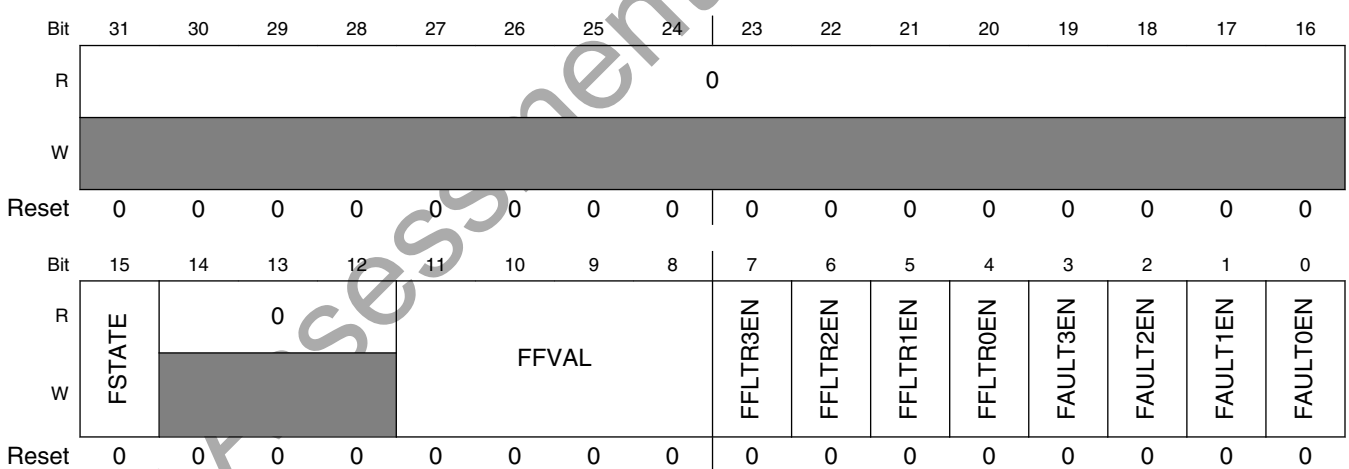
FTM_FILTER field descriptions (continued)

Field	Description
11–8 CH2FVAL	Channel 2 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
7–4 CH1FVAL	Channel 1 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
CH0FVAL	Channel 0 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.

38.4.20 Fault Control (FTM_FLTCTRL)

This register selects the filter value for the fault inputs, enables the fault inputs and the fault inputs filter. This register also controls the output state when a fault event happens.

Address: 0h base + 7Ch offset = 7Ch



FTM_FLTCTRL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 FSTATE	Fault output state This configuration allows to put the FTM outputs tri-stated when a fault event is ongoing. This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_FLTCTRL field descriptions (continued)

Field	Description
	0 FTM outputs will be placed into safe values when fault events in ongoing (defined by POL bits). 1 FTM outputs will be tri-stated when fault event is ongoing
14–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 FFVAL	Fault Input Filter Selects the filter value for the fault inputs. The fault filter is disabled when the value is zero. **NOTE:** Writing to this field has immediate effect and must be done only when the fault control or all fault inputs are disabled. Failure to do this could result in a missing fault detection.
7 FFLTR3EN	Fault Input 3 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
6 FFLTR2EN	Fault Input 2 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
5 FFLTR1EN	Fault Input 1 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
4 FFLTR0EN	Fault Input 0 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
3 FAULT3EN	Fault Input 3 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.
2 FAULT2EN	Fault Input 2 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_FLTCTRL field descriptions (continued)

Field	Description
	0 Fault input is disabled. 1 Fault input is enabled.
1 FAULT1EN	Fault Input 1 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.
0 FAULT0EN	Fault Input 0 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.

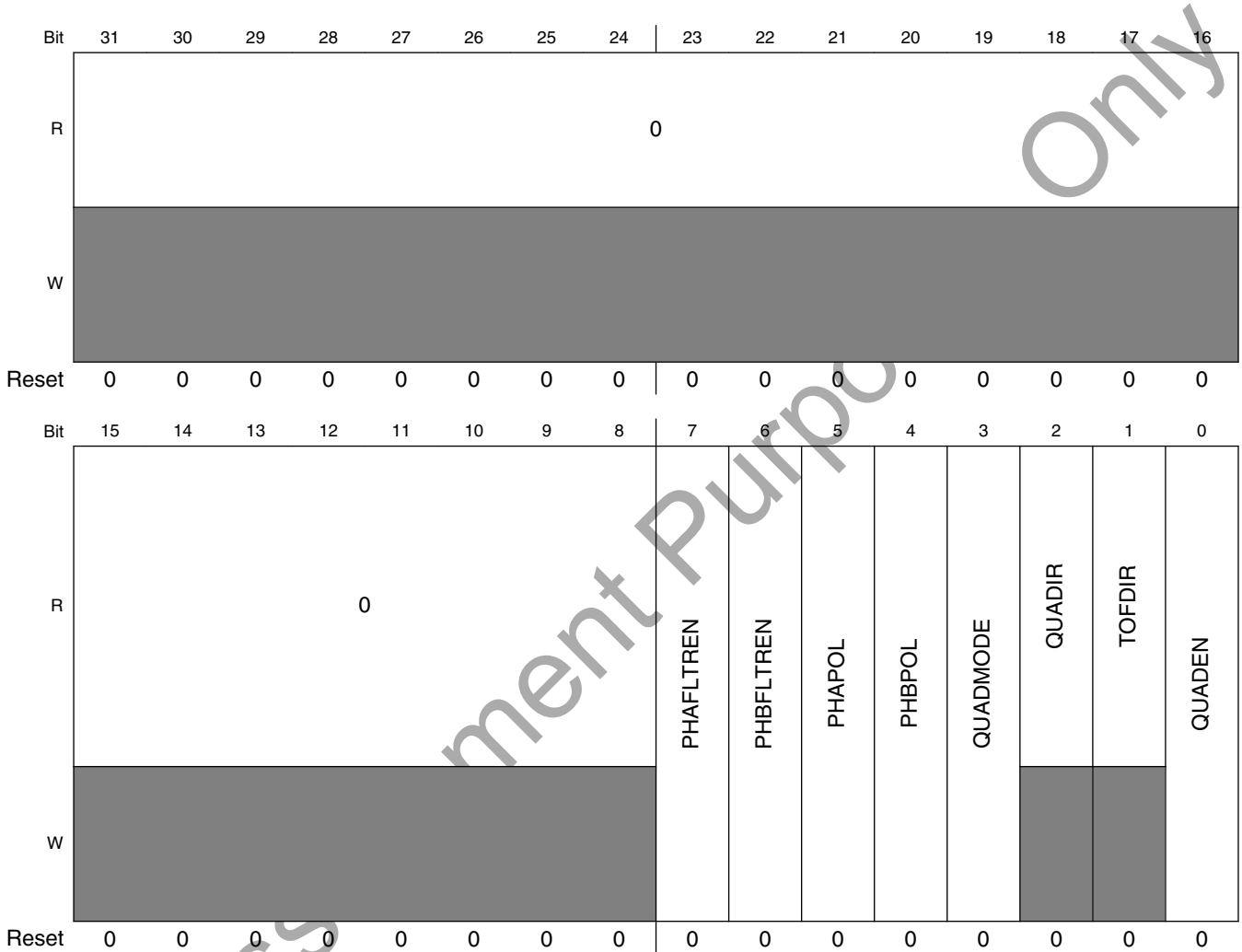
For Assessment Purpo

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.4.21 Quadrature Decoder Control And Status (FTM_QDCTRL)

This register has the control and status bits for the Quadrature Decoder mode.

Address: 0h base + 80h offset = 80h



ssment Purpo Only

FTM_QDCTRL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 PHAFLTREN	Phase A Input Filter Enable Enables the filter for the quadrature decoder phase A input. The filter value for the phase A input is defined by the CH0FVAL field of FILTER. The phase A filter is also disabled when CH0FVAL is zero. 0 Phase A input filter is disabled. 1 Phase A input filter is enabled.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_QDCTRL field descriptions (continued)

Field	Description
6 PHBFLTREN	Phase B Input Filter Enable Enables the filter for the quadrature decoder phase B input. The filter value for the phase B input is defined by the CH1FVAL field of FILTER. The phase B filter is also disabled when CH1FVAL is zero. 0 Phase B input filter is disabled. 1 Phase B input filter is enabled.
5 PHAPOL	Phase A Input Polarity Selects the polarity for the quadrature decoder phase A input. 0 Normal polarity. Phase A input signal is not inverted before identifying the rising and falling edges of this signal. 1 Inverted polarity. Phase A input signal is inverted before identifying the rising and falling edges of this signal.
4 PHBPOL	Phase B Input Polarity Selects the polarity for the quadrature decoder phase B input. 0 Normal polarity. Phase B input signal is not inverted before identifying the rising and falling edges of this signal. 1 Inverted polarity. Phase B input signal is inverted before identifying the rising and falling edges of this signal.
3 QUADMODE	Quadrature Decoder Mode Selects the encoding mode used in the Quadrature Decoder mode. 0 Phase A and phase B encoding mode. 1 Count and direction encoding mode.
2 QUADIR	FTM Counter Direction In Quadrature Decoder Mode Indicates the counting direction. 0 Counting direction is decreasing (FTM counter decrement). 1 Counting direction is increasing (FTM counter increment).
1 TOFDIR	Timer Overflow Direction In Quadrature Decoder Mode 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 (CNTIN register) 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 (CNTIN register).
0 QUADEN	Quadrature Decoder Mode Enable Enables the Quadrature Decoder mode. In this mode, the phase A and B input signals control the FTM counter direction. The Quadrature Decoder mode has precedence over the other modes. See Channel Modes. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Quadrature Decoder mode is disabled. 1 Quadrature Decoder mode is enabled.

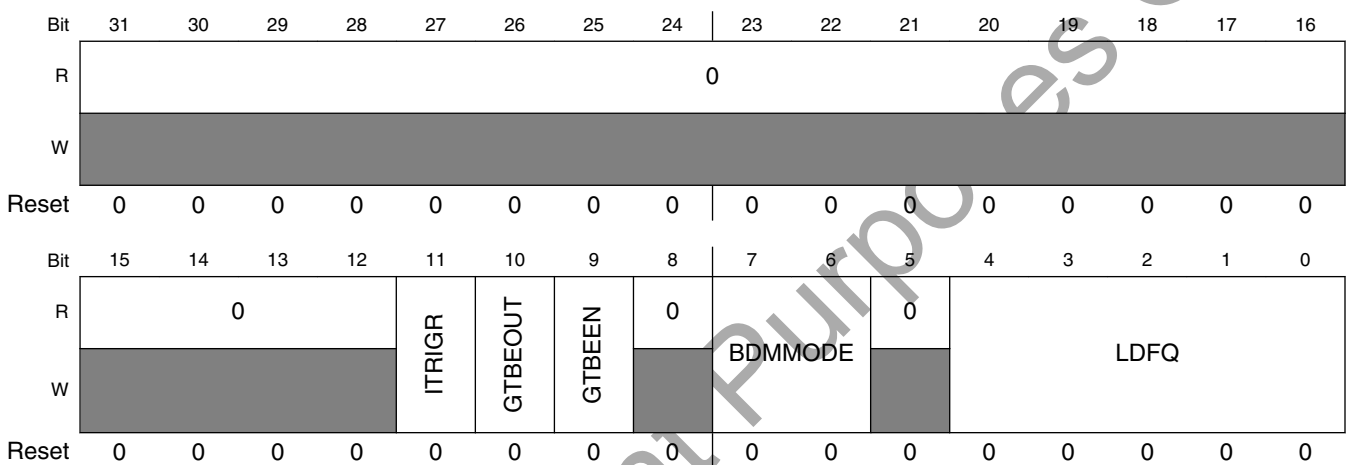
Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.4.22 Configuration (FTM_CONF)

This register selects the number of times that a reload opportunity should occur before the RF bit is set, the FTM behavior in Debug modes, the use of an external global time base, and the global time base signal generation.

This register also controls if initialization trigger should be generated when a reload point is reached.

Address: 0h base + 84h offset = 84h



FTM_CONF field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 ITRIGR	Initialization trigger on Reload Point This bit controls whether an initialization trigger is generated when a reload point configured by PWMLOAD register is reached considering the FTM_CONF[LDFQ] settings. 0 Initialization trigger is generated on counter wrap events. 1 Initialization trigger is generated when a reload point is reached.
10 GTBEOUT	Global Time Base Output Enables the global time base signal generation to other FTMs. 0 A global time base signal generation is disabled. 1 A global time base signal generation is enabled.
9 GTBEEN	Global Time Base Enable Configures the FTM to use an external global time base signal that is generated by another FTM. 0 Use of an external global time base is disabled. 1 Use of an external global time base is enabled.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

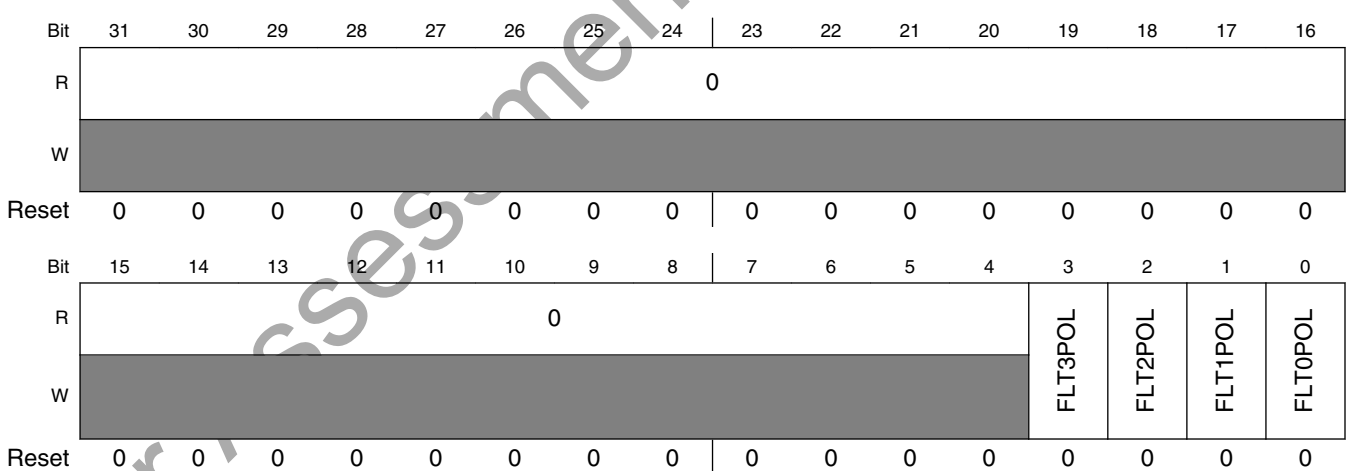
FTM_CONF field descriptions (continued)

Field	Description
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 BDMMODE	Debug Mode Selects the FTM behavior in Debug mode. See Debug mode.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LDFQ	Load Frequency Selects PWM reload frequency. LDFQ = 0: RF bit is set every reload oportunity. LDFQ = 1: RF bit is set every 2 reload oportunities. LDFQ = 2: RF bit is set every 3 reload oportunities. LDFQ = 3: RF bit is set every 4 reload oportunities. This pattern continues up to a maximum of 32.

38.4.23 FTM Fault Input Polarity (FTM_FLTPOL)

This register defines the fault inputs polarity.

Address: 0h base + 88h offset = 88h



FTM_FLTPOL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 FLT3POL	Fault Input 3 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_FLTPOL field descriptions (continued)

Field	Description
	0 The fault input polarity is active high. A 1 at the fault input indicates a fault. 1 The fault input polarity is active low. A 0 at the fault input indicates a fault.
2 FLT2POL	Fault Input 2 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A 1 at the fault input indicates a fault. 1 The fault input polarity is active low. A 0 at the fault input indicates a fault.
1 FLT1POL	Fault Input 1 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A 1 at the fault input indicates a fault. 1 The fault input polarity is active low. A 0 at the fault input indicates a fault.
0 FLT0POL	Fault Input 0 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A 1 at the fault input indicates a fault. 1 The fault input polarity is active low. A 0 at the fault input indicates a fault.

38.4.24 Synchronization Configuration (FTM_SYNCONF)

This register selects the PWM synchronization configuration, SWOCTRL, INVCTRL and CNTIN registers synchronization, if FTM clears the TRIGj bit, where j = 0, 1, 2, when the hardware trigger j is detected.

Address: 0h base + 8Ch offset = 8Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											HWSOC	HWINVC	HWOM	HWWRBUF	HWRSTCNT
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			SWSOC	SWINVC	SWOM	SWWRBUF	SWRSTCNT	SYNCMODE	0	SWOC	INVC	0	CNTINC	0	HWTRIGMODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FC Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_SYNCONF field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 HWSOC	Software output control synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the SWOCTRL register synchronization. 1 A hardware trigger activates the SWOCTRL register synchronization.
19 HWINVC	Inverting control synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the INVCTRL register synchronization. 1 A hardware trigger activates the INVCTRL register synchronization.
18 HWOM	Output mask synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the OUTMASK register synchronization. 1 A hardware trigger activates the OUTMASK register synchronization.
17 HWWRBUF	MOD, HCR, CNTIN, and CV registers synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate MOD, HCR, CNTIN, and CV registers synchronization. 1 A hardware trigger activates MOD, HCR, CNTIN, and CV registers synchronization.
16 HWRSTCNT	FTM counter synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the FTM counter synchronization. 1 A hardware trigger activates the FTM counter synchronization.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 SWSOC	Software output control synchronization is activated by the software trigger. 0 The software trigger does not activate the SWOCTRL register synchronization. 1 The software trigger activates the SWOCTRL register synchronization.
11 SWINVC	Inverting control synchronization is activated by the software trigger. 0 The software trigger does not activate the INVCTRL register synchronization. 1 The software trigger activates the INVCTRL register synchronization.
10 SWOM	Output mask synchronization is activated by the software trigger. 0 The software trigger does not activate the OUTMASK register synchronization. 1 The software trigger activates the OUTMASK register synchronization.
9 SWWRBUF	MOD, HCR, CNTIN, and CV registers synchronization is activated by the software trigger. 0 The software trigger does not activate MOD, HCR, CNTIN, and CV registers synchronization. 1 The software trigger activates MOD, HCR, CNTIN, and CV registers synchronization.
8 SWRSTCNT	FTM counter synchronization is activated by the software trigger. 0 The software trigger does not activate the FTM counter synchronization. 1 The software trigger activates the FTM counter synchronization.
7 SYNCMODE	Synchronization Mode Selects the PWM Synchronization mode.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_SYNCONF field descriptions (continued)

Field	Description
	0 Legacy PWM synchronization is selected. 1 Enhanced PWM synchronization is selected.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 SWOC	SWOCTRL Register Synchronization 0 SWOCTRL register is updated with its buffer value at all rising edges of system clock. 1 SWOCTRL register is updated with its buffer value by the PWM synchronization.
4 INVC	INVCTRL Register Synchronization 0 INVCTRL register is updated with its buffer value at all rising edges of system clock. 1 INVCTRL register is updated with its buffer value by the PWM synchronization.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CNTINC	CNTIN Register Synchronization 0 CNTIN register is updated with its buffer value at all rising edges of system clock. 1 CNTIN register is updated with its buffer value by the PWM synchronization.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 HWTRIGMODE	Hardware Trigger Mode 0 FTM clears the TRIGj bit when the hardware trigger j is detected, where j = 0, 1,2. 1 FTM does not clear the TRIGj bit when the hardware trigger j is detected, where j = 0, 1,2.

For Assessment

38.4.25 FTM Inverting Control (FTM_INVCTRL)

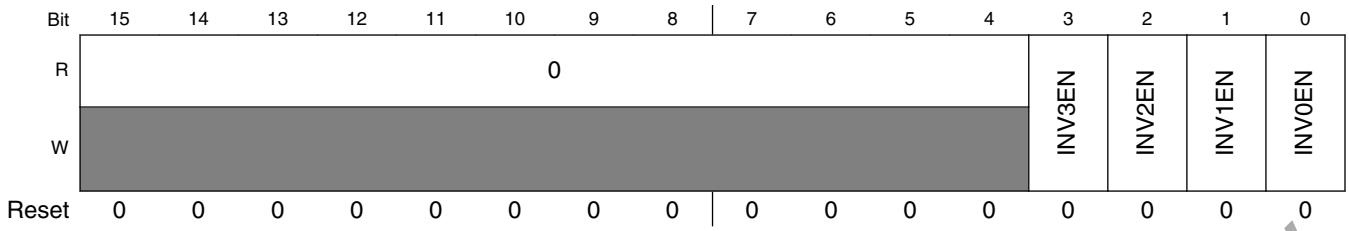
This register controls when the channel (n) output becomes the channel (n+1) output, and channel (n+1) output becomes the channel (n) output. Each INVmEN bit enables the inverting operation for the corresponding pair channels m.

This register has a write buffer. The INVmEN bit is updated by the INVCTRL register synchronization.

Address: 0h base + 90h offset = 90h

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required



FTM_INVCTRL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 INV3EN	Pair Channels 3 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
2 INV2EN	Pair Channels 2 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
1 INV1EN	Pair Channels 1 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
0 INV0EN	Pair Channels 0 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.

For Assessme

38.4.26 FTM Software Output Control (FTM_SWOCTRL)

This register enables software control of channel (n) output and defines the value forced to the channel (n) output:

- The CHnOC bits enable the control of the corresponding channel (n) output by software.
- The CHnOCV bits select the value that is forced at the corresponding channel (n) output.

This register has a write buffer. The fields are updated by the SWOCTRL register synchronization.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + 94h offset = 94h

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	CH7OCV	CH6OCV	CH5OCV	CH4OCV	CH3OCV	CH2OCV	CH1OCV	CH0OCV	CH7OC	CH6OC	CH5OC	CH4OC	CH3OC	CH2OC	CH1OC	CH0OC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTM_SWOCTRL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 CH7OCV	Channel 7 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
14 CH6OCV	Channel 6 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
13 CH5OCV	Channel 5 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
12 CH4OCV	Channel 4 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
11 CH3OCV	Channel 3 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
10 CH2OCV	Channel 2 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
9 CH1OCV	Channel 1 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
8 CH0OCV	Channel 0 Software Output Control Value

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_SWOCTRL field descriptions (continued)

Field	Description
	0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
7 CH7OC	Channel 7 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
6 CH6OC	Channel 6 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
5 CH5OC	Channel 5 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
4 CH4OC	Channel 4 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
3 CH3OC	Channel 3 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
2 CH2OC	Channel 2 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
1 CH1OC	Channel 1 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
0 CH0OC	Channel 0 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.

For Ass

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.4.27 FTM PWM Load (FTM_PWMLOAD)

Enables the reload of the MOD, HCR, CNTIN, C(n)V, and C(n+1)V registers with the values of their write buffers when the FTM counter changes from the MOD register value to its next value or when a channel (j) match occurs. A match occurs for channel (j) when FTM counter = C(j)V. A reload can also occurs when FTM counter = HCR register at a half cycle match. This register also controls the local and global load mechanisms.

Address: 0h base + 98h offset = 98h

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	GLEN	LDOK	HCSEL	CH7SEL	CH6SEL	CH5SEL	CH4SEL	CH3SEL	CH2SEL	CH1SEL	CH0SEL
W					GLDOK											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTM_PWMLOAD field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 GLDOK	Global Load OK This bit controls the global load mechanism. It generates a pulse at FTM module global load output with one FTM clock cycle width, which is used to set LDOK bits of FTM and other modules (including other FTMs). This bit is self-cleared and read value is always zero. The global load mechanism depends on SoC specific information. Refer to FTM SoC specific information to more details. 0 No action. 1 LDOK bit is set.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_PWMLOAD field descriptions (continued)

Field	Description
10 GLEN	Global Load Enable This bit enables the global load mechanism implemented by GLDOK. If GLEN bit is set, then an external event on the FTM global load input sets the LDOK bit. The clear of the LDOK bit is done by CPU writes '0' to the bit. 0 Global Load Ok disabled. 1 Global Load OK enabled. A pulse event on the module global load input sets the LDOK bit.
9 LDOK	Load Enable Enables the loading of the MOD, CNTIN, HCR and CV registers with the values of their buffers. The LDOK bit can also be set by the Global Load mechanism if GLEN bit is enabled. 0 Loading updated values is disabled. 1 Loading updated values is enabled.
8 HCSEL	Half Cycle Select This bit enables the half cycle match as a reload oportunity. A half cycle is defined by when the FTM counter matches the HCR register. 0 Half cycle reload is disabled and it is not considered as a reload opportunity. 1 Half cycle reload is enabled and it is considered as a reload opportunity.
7 CH7SEL	Channel 7 Select 0 Channel match is not included as a reload opportunity. 1 Channel match is included as a reload opportunity.
6 CH6SEL	Channel 6 Select 0 Channel match is not included as a reload opportunity. 1 Channel match is included as a reload opportunity.
5 CH5SEL	Channel 5 Select 0 Channel match is not included as a reload opportunity. 1 Channel match is included as a reload opportunity.
4 CH4SEL	Channel 4 Select 0 Channel match is not included as a reload opportunity. 1 Channel match is included as a reload opportunity.
3 CH3SEL	Channel 3 Select 0 Channel match is not included as a reload opportunity. 1 Channel match is included as a reload opportunity.
2 CH2SEL	Channel 2 Select 0 Channel match is not included as a reload opportunity. 1 Channel match is included as a reload opportunity.
1 CH1SEL	Channel 1 Select 0 Channel match is not included as a reload opportunity. 1 Channel match is included as a reload opportunity.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

FTM_PWMLOAD field descriptions (continued)

Field	Description
0 CH0SEL	Channel 0 Select 0 Channel match is not included as a reload opportunity. 1 Channel match is included as a reload opportunity.

38.4.28 Half Cycle Register (FTM_HCR)

The Half Cycle Register contains the match value for FTM half cycle reload feature. After FTM counter reaches this value, a reload opportunity is generated if FTM_PWMLOAD[HCSEL] is enabled.

Writing to the HCR register latches the value into a buffer. The HCR register is updated with the value of its write buffer according to Registers updated from write buffers.

Address: 0h base + 9Ch offset = 9Ch

Bit	31	30	29	28	27	26	25	24	23	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																HCVAL															
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

FTM_HCR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
HCVAL	Half Cycle Value

38.5 Functional description

The notation used in this document to represent the counters and the generation of the signals is shown in the following figure.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

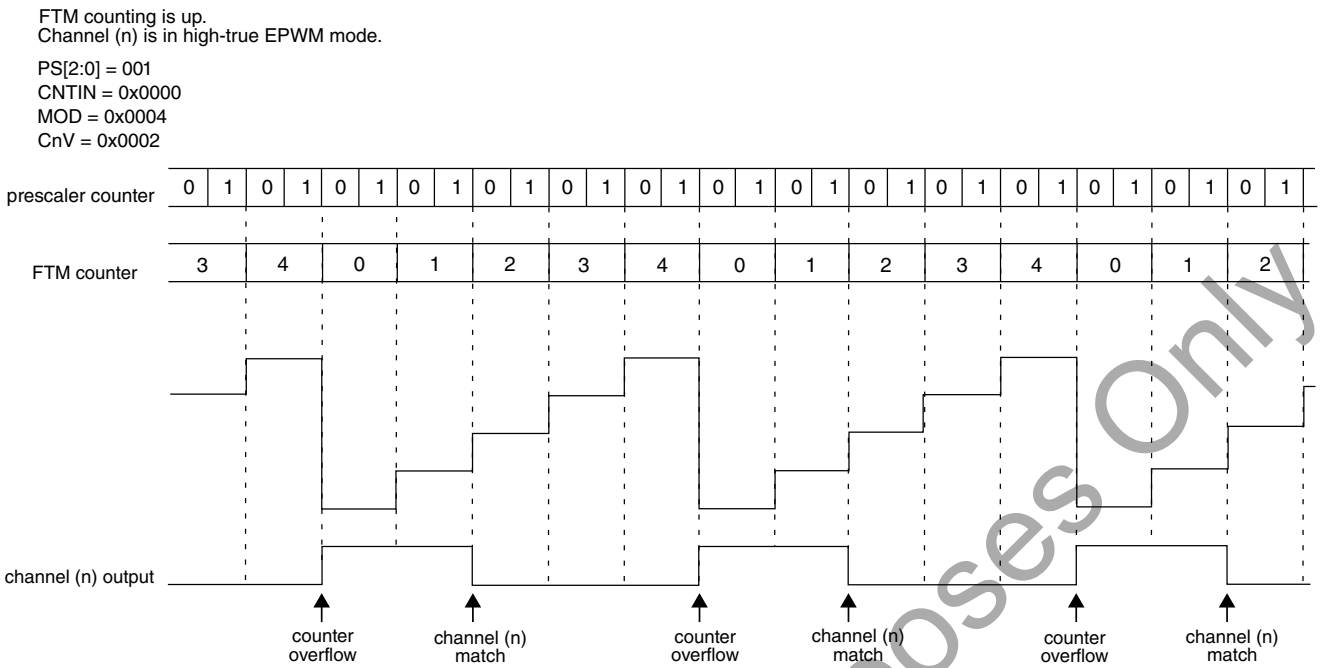


Figure 38-4. Notation used

38.5.1 Clock source

The FTM has only one clock domain: the system clock.

38.5.1.1 Counter clock source

The CLKS[1:0] bits in the SC register select one of three possible clock sources for the FTM counter or disable the FTM counter. After any chip reset, CLKS[1:0] = 0:0 so no clock source is selected.

The CLKS[1:0] bits may be read or written at any time. Disabling the FTM counter by writing 0:0 to the CLKS[1:0] bits does not affect the FTM counter value or other registers.

The fixed frequency clock is an alternative clock source for the FTM counter that allows the selection of a clock other than the system clock or an external clock. This clock input is defined by chip integration. Refer to the chip specific documentation for further information. Due to FTM hardware implementation limitations, the frequency of the fixed frequency clock must not exceed 1/2 of the system clock frequency.

For Assessment Purposes Only

The external clock passes through a synchronizer clocked by the system clock to assure that counter transitions are properly aligned to system clock transitions.Therefore, to meet Nyquist criteria considering also jitter, the frequency of the external clock source must not exceed 1/4 of the system clock frequency.

38.5.2 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 FTM counter.

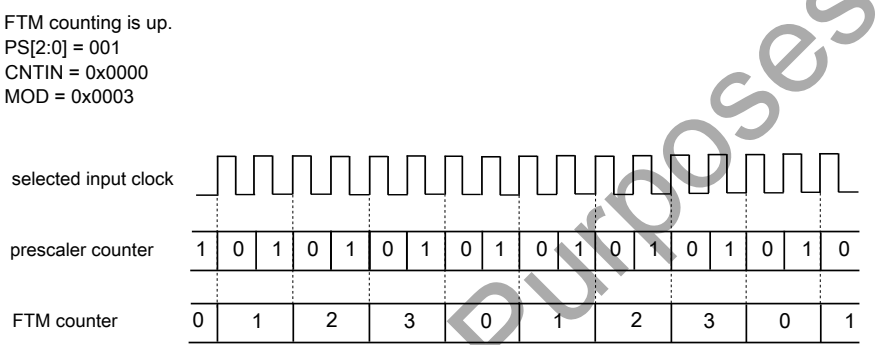


Figure 38-5. Example of the prescaler counter

38.5.3 Counter

The FTM has a 16-bit counter that is used by the channels either for input or output modes. The FTM counter clock is the selected clock divided by the prescaler.

The FTM counter has these modes of operation:

- Up counting
- Up-down counting
- Quadrature Decoder mode

38.5.3.1 Up counting

Up counting is selected when:

- QUADEN = 0, and
- CPWMS = 0

For Assessment Purposes Only

CNTIN defines the starting value of the count and MOD defines the final value of the count, see the following figure. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is reloaded with the value of CNTIN.

The FTM period when using up counting is (MOD – CNTIN + 0x0001) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to CNTIN.

A counter event happens at the same time of TOF bit set when the FTM counter changes from MOD to CNTIN. See Counter events for more details.

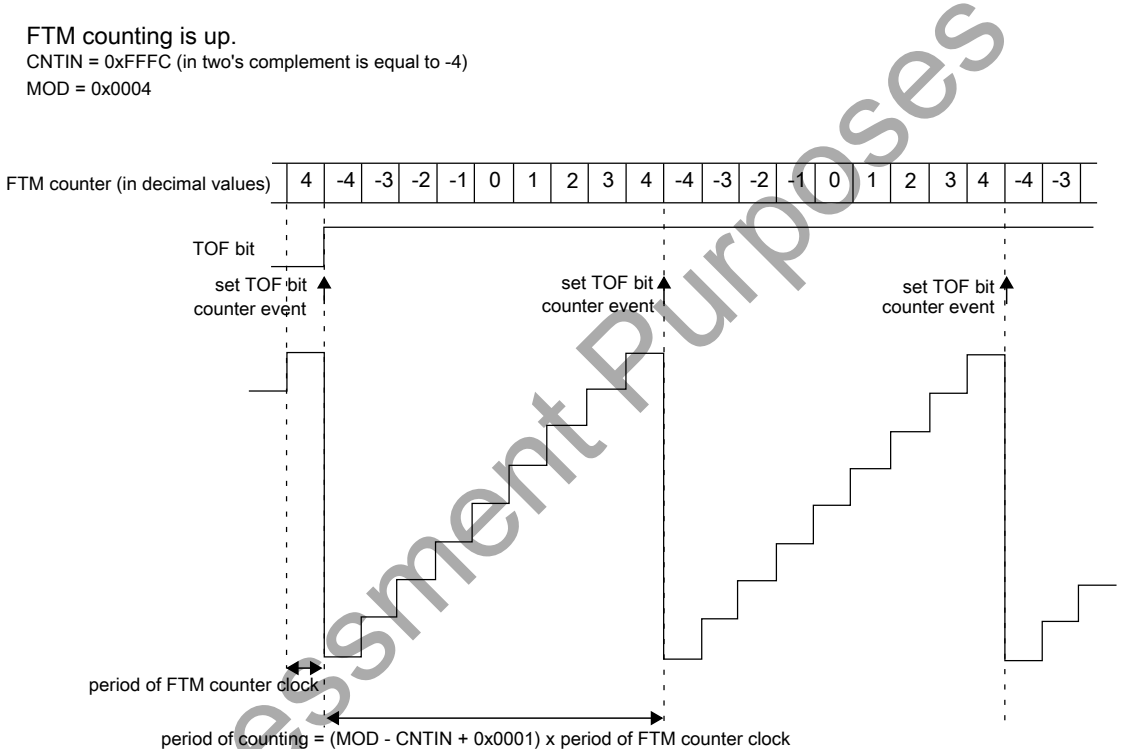


Figure 38-6. Example of FTM up and signed counting

Table 38-3. FTM counting based on CNTIN value

When	Then
CNTIN = 0x0000	The FTM counting is equivalent to TPM up counting, that is, up and unsigned counting. See the following figure.
CNTIN[15] = 1	The initial value of the FTM counter is a negative number in two's complement, so the FTM counting is up and signed.
CNTIN[15] = 0 and CNTIN ≠ 0x0000	The initial value of the FTM counter is a positive number, so the FTM counting is up and unsigned.

Assessment Purposes Only

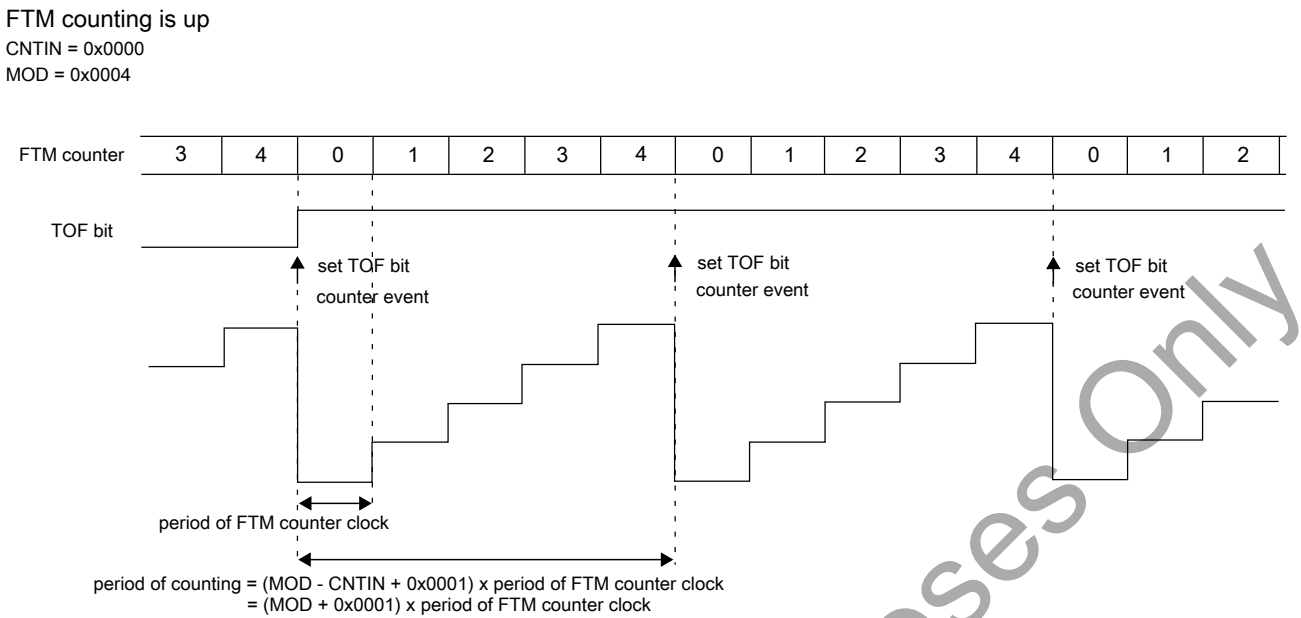


Figure 38-7. Example of FTM up counting with CNTIN = 0x0000

Note

- FTM operation is only valid when the value of the CNTIN register is less than the value of the MOD register, either in the unsigned counting or signed counting. It is the responsibility of the software to ensure that the values in the CNTIN and MOD registers meet this requirement. Any values of CNTIN and MOD that do not satisfy this criteria can result in unpredictable behavior.
- MOD = CNTIN is a redundant condition. In this case, the FTM counter is always equal to MOD and the TOF bit is set in each rising edge of the FTM counter clock.
- When MOD = 0x0000, CNTIN = 0x0000, for example after reset, and FTMEN = 1, the FTM counter remains stopped at 0x0000 until a non-zero value is written into the MOD or CNTIN registers.
- Setting CNTIN to be greater than the value of MOD is not recommended as this unusual setting may make the FTM operation difficult to comprehend. However, there is no restriction on this configuration, and an example is shown in the following figure.

For Assessment Purposes Only

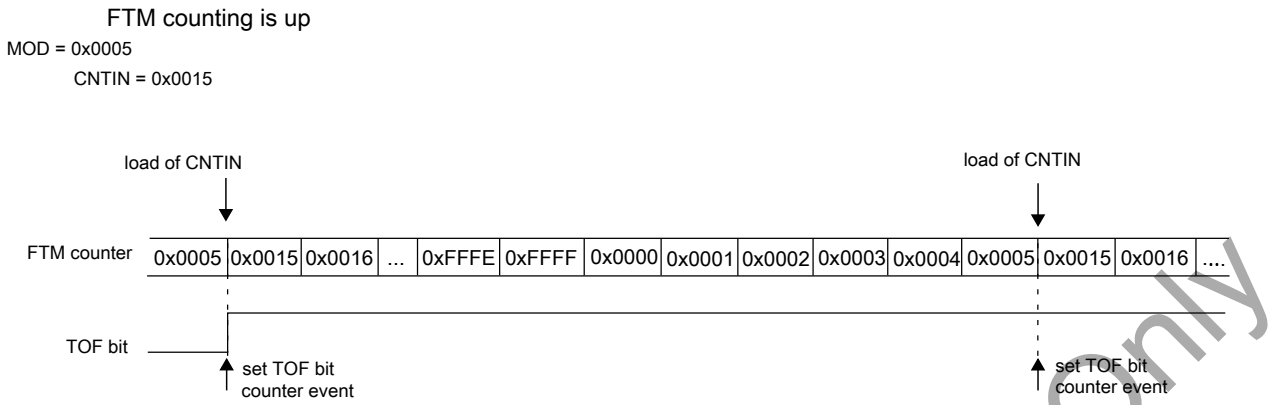


Figure 38-8. Example of up counting when the value of CNTIN is greater than the value of MOD

38.5.3.2 Up-down counting

Up-down counting is selected when:

- QUADEN = 0, and
- CPWMS = 1

CNTIN defines the starting value of the count and MOD defines the final value of the count. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is decremented until it returns to the value of CNTIN and the up-down counting restarts.

The FTM period when using up-down counting is 2 × (MOD – CNTIN) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to (MOD – 1).

If (CNTIN = 0x0000), the FTM counting is equivalent to TPM up-down counting, that is, up-down and unsigned counting. See the following figure.

For Assessment Purposes Only

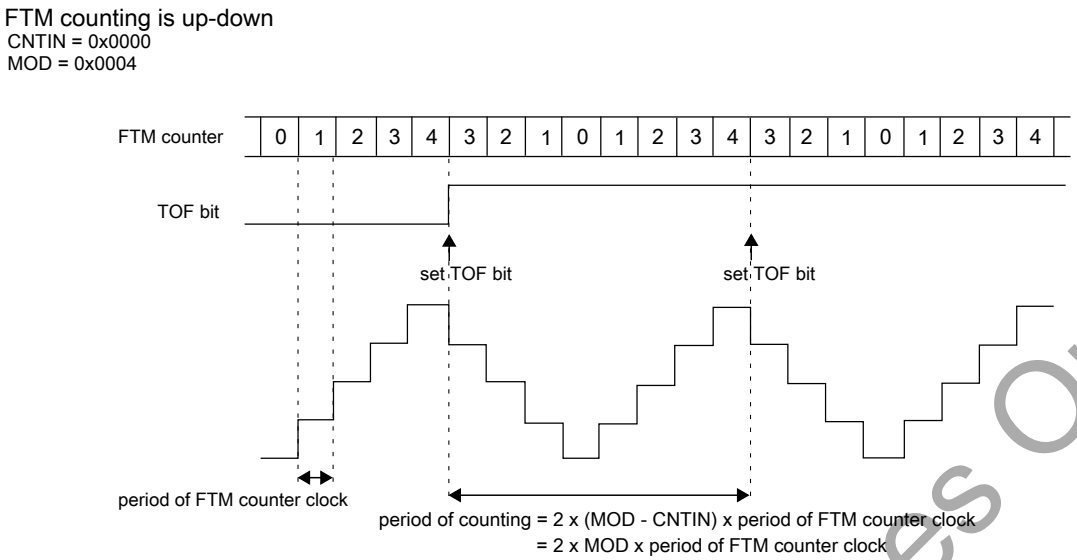


Figure 38-9. Example of up-down counting when CNTIN = 0x0000

Note

When CNTIN is different from zero in the up-down counting, a valid CPWM signal is generated:

- if CnV > CNTIN, or
- if CnV = 0 or if CnV[15] = 1. In this case, 0% CPWM is generated.

The figure below shows the possible counter events when in up-down counting mode. See Counter events for more details.

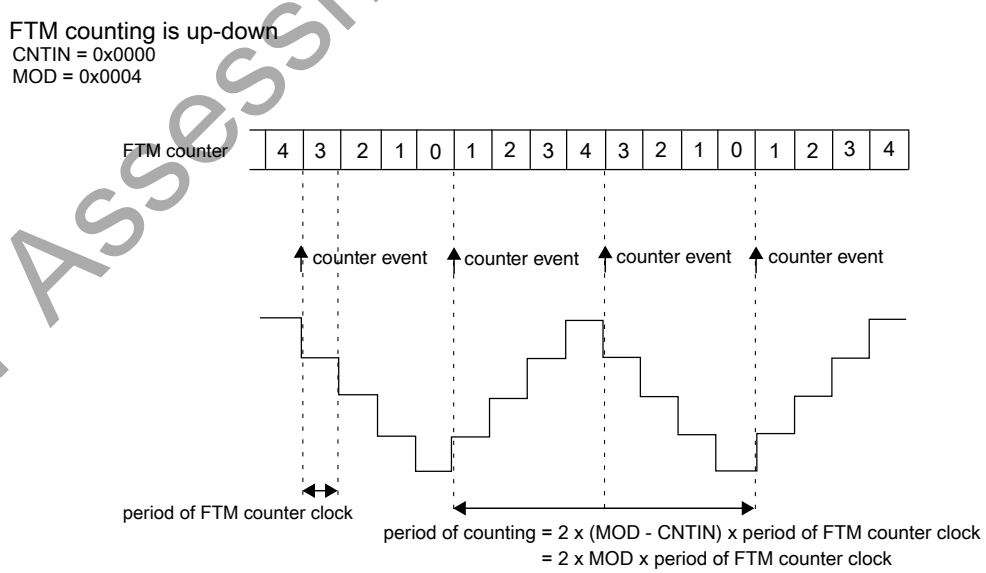


Figure 38-10. Example of counter events in up-down counting mode when CNTIN = 0x0000

For Assessment Purposes Only

38.5.3.3 Free running counter

If (FTMEN = 0) and (MOD = 0x0000 or MOD = 0xFFFF), the FTM counter is a free running counter. In this case, the FTM counter runs free from 0x0000 through 0xFFFF and the TOF bit is set when the FTM counter changes from 0xFFFF to 0x0000. See the following figure.

A counter event occurs at the same time of TOF bit set when the FTM counter changes from 0xFFFF to 0x0000. See Counter events for more details.

Figure 38-11. Example when the FTM counter is free running

The FTM counter is also a free running counter when:

- FTMEN = 1
- QUADEN = 0
- CPWMS = 0
- CNTIN = 0x0000, and
- MOD = 0xFFFF

38.5.3.4 Counter reset

Any one of the following cases resets the FTM counter to the value in the CNTIN register and the channels output to its initial value, except for channels in Output Compare mode.

- Any write to CNT.
- FTM counter synchronization.
- A channel in Input Capture mode with ICRST = 1 (FTM Counter Reset in Input Capture Mode).

Note that reseting the counter also generates a counter event. See Counter events for more details.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.5.3.5 Counter events

Counter events can be used as reload opportunities to FTM register sychronization mechanism. See Half and full cycle reload for more details. There are some possible counter events depending on the counter mode. Please see the table below for more details.

Table 38-4. FTM counter events

When	Then
FTM counter is in up counting mode or freerunning	• A counter event happens at the same time of TOF bit set when the FTM counter changes from MOD to CNTIN (counter wrap). Figure at Up counting shows the counter event generation. • When in freerunning, there is a counter event when FTM counter changes from 0xFFFF to 0x0000. Figure at Free running counter shows the counter event generation.
FTM counter is in up-down counting mode	• In up-down counting mode, there are two possible counter events when FTM counter turns from down to up counting and when counter turns from up to down counting. User can select which point will be used to generate the counter event. Figure at Up-down counting shows the possible counter events.
FTM counter is reseted (see Counter reset) or a value different from zero is written at CLKS field	• In up-counting mode, all counter reset events or a write in the CLKS with a value different from zero generates a counter event. • In up-down counting mode, counter reset events only generates a counter event if the minimum load point when FTM counter turns from down to up counting is configured. A write in the CLKS with a value different from zero always generates a counter event in up-down counting mode.

Assess

38.5.4 Channel Modes

The following table shows the channel modes selection.

Table 38-5. Channel Modes Selection

DECAPEN	MCOMBINE	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	X	X	X	XX	00	Pin not used for FTM—revert the channel pin to general purpose I/O or other peripheral control	
0	0	0	0	00	01	Input Capture	Capture on Rising Edge Only

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 38-5. Channel Modes Selection (continued)

DECAPEN	MCOMBINE	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
					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
				1X	10	Edge-Aligned PWM	High-true pulses (clear Output on match)
					X1		Low-true pulses (set Output on match)
			1	XX	10	Center-Aligned PWM	High-true pulses (clear Output on match-up)
					X1		Low-true pulses (set Output on match-up)
		1	0	XX	10	Combine PWM	High-true pulses (set on channel (n) match, and clear on channel (n+1) match)
					X1		Low-true pulses (clear on channel (n) match, and set on channel (n+1) match)
	1	0	X	XX	XX	Reserved for future use	
	1	1	0	XX	10	Modified Combine PWM	High-true pulses (set on channel (n) match, and

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 38-5. Channel Modes Selection (continued)

DECAPEN	MCOMBINE	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
							clear on channel (n+1) match)
					X1		Low-true pulses (clear on channel (n) match, and set on channel (n+1) match)
1	0	0	0	X0	See Table 38-6.	Dual Edge Capture	One-Shot Capture mode
				X1			Continuous Capture mode

Table 38-6. Dual Edge Capture Mode — Edge Polarity Selection

ELSnB	ELSnA	Channel Port Enable	Detected Edges
0	0	Disabled	No edge
0	1	Enabled	Rising edge
1	0	Enabled	Falling edge
1	1	Enabled	Rising and falling edges

38.5.5 Input Capture mode

The Input Capture mode is selected when:

- DECAPEN = 0
- MCOMBINE = 0
- COMBINE = 0
- 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 FTM 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.

For Assessment Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

When a channel is configured for input capture, the FTMxCHn 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 system clock divided by 4, which is required to meet Nyquist criteria for signal sampling.

Writes to the CnV register is ignored in Input Capture mode.

While in Debug mode, the input capture function works as configured. When a selected edge event occurs, the FTM counter value, which is frozen because of Debug, is captured into the CnV register and the CHnF bit is set.

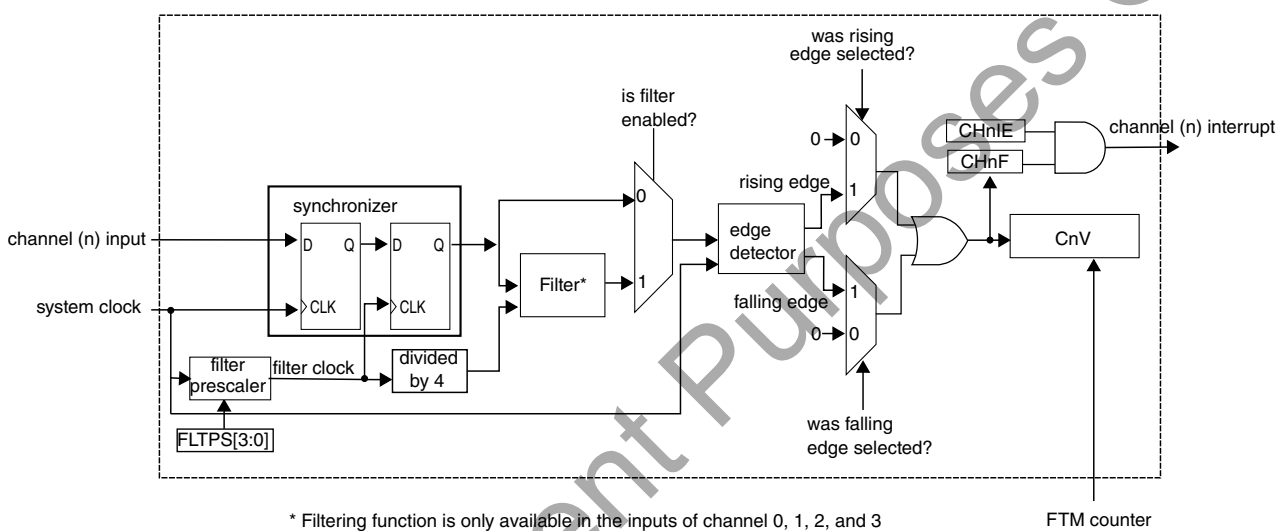


Figure 38-12. Input Capture mode

If the channel input does not have a filter enabled, then the input signal is always delayed 3 rising edges of the system clock, that is, two rising edges to the synchronizer plus one more rising edge to the edge detector. In other words, the CHnF bit is set on the third rising edge of the system clock after a valid edge occurs on the channel input.

38.5.5.1 Filter for Input Capture mode

The filter function is only available on channels 0, 1, 2, and 3.

First, the input signal is synchronized by the system clock. Following synchronization, the input signal enters the filter block. See the following figure.

For Assessment Purposes Only

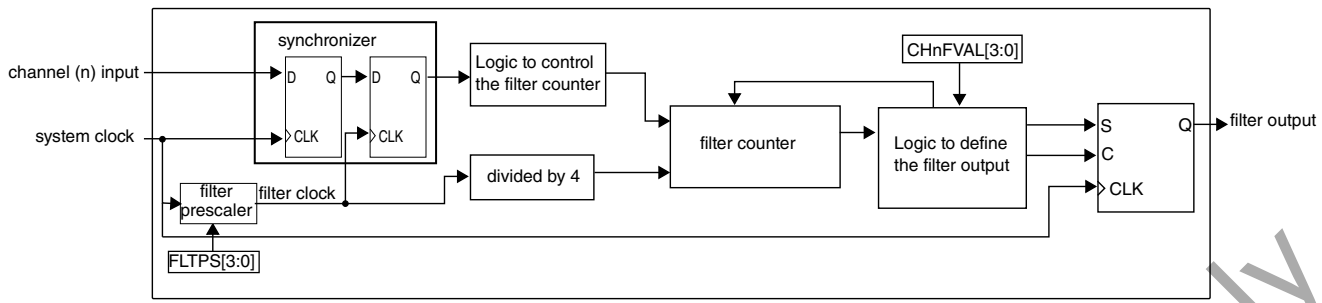


Figure 38-13. Channel input filter

NOTE

The Channel Input Filter internal counter clock is further divided by 4 in order to reject high frequency glitches.

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], the state change of the input signal is validated. It is then transmitted as a pulse edge to the edge detector.

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 filter clock cycles) is regarded as a glitch and is not passed on to the edge detector. 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 3 rising edges (1 system clock + 2 filter clocks). If (CHnFVAL[3:0] ≠ 0000), then the delay through the filter logic is given by the expression: 2 system clocks + (2 + 4 x CHnFVAL) filter clocks. In other words, CHnF is set (2 system clocks + 2 filter clocks + 4 filter clocks × CHnFVAL[3:0]) clock periods after a valid edge occurs on the channel input.

The clock for the counter in the channel input filter is the filter clock divided by 4.

For Assessment Purposes Only

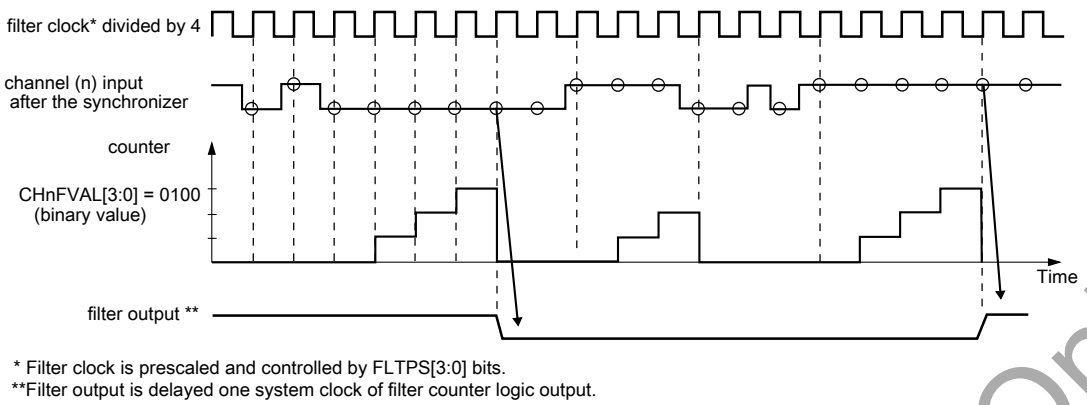


Figure 38-14. Channel input filter example

The figure below shows the delay through the input filter logic considering each internal filter element. In this example the filter prescaler is set to zero. Note that the input signal is delayed only by the synchronizer and edge dector logic if the filter is disabled.

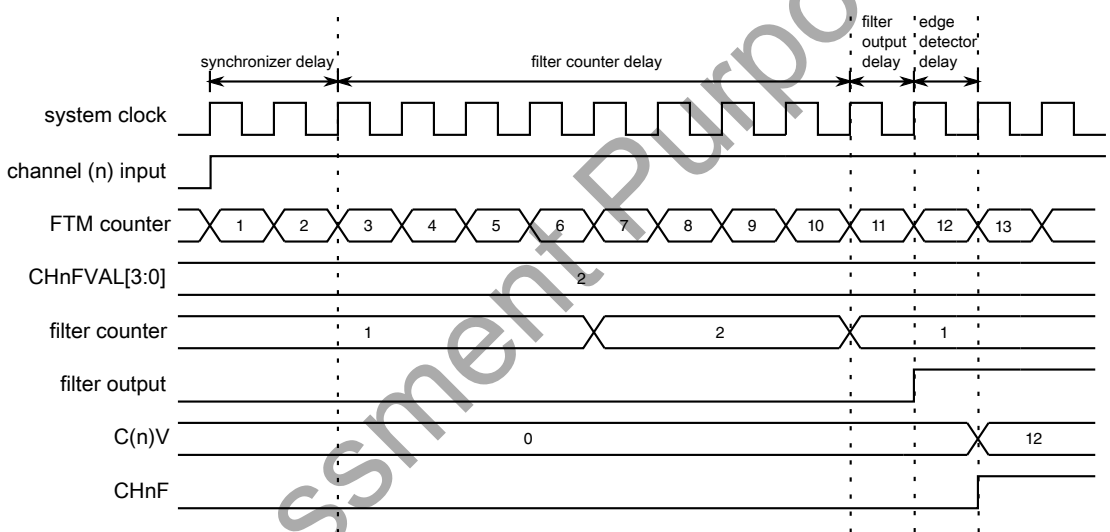


Figure 38-15. Input capture example

38.5.5.2 FTM Counter Reset in Input Capture Mode

If the channel (n) is in input capture mode and CnSC[ICRST = 1], then when the selected input capture event occurs in the channel (n) input signal, the current value of the FTM counter is captured into the CnV register, the CHnF bit is set, the channel (n) interrupt is generated (if CHnIE = 1) and the FTM counter is reset to the CNTIN register value.

This allows the FTM to measure a period/pulse being applied to FTM_CHn (counts of the FTM clock input) without having to implement a subtraction calculation in software subsequent to the event occurring.

For Assessment Purposes Only

The figure below shows the FTM counter reset when the selected input capture event is detected in a channel in input capture mode with ICRST = 1.

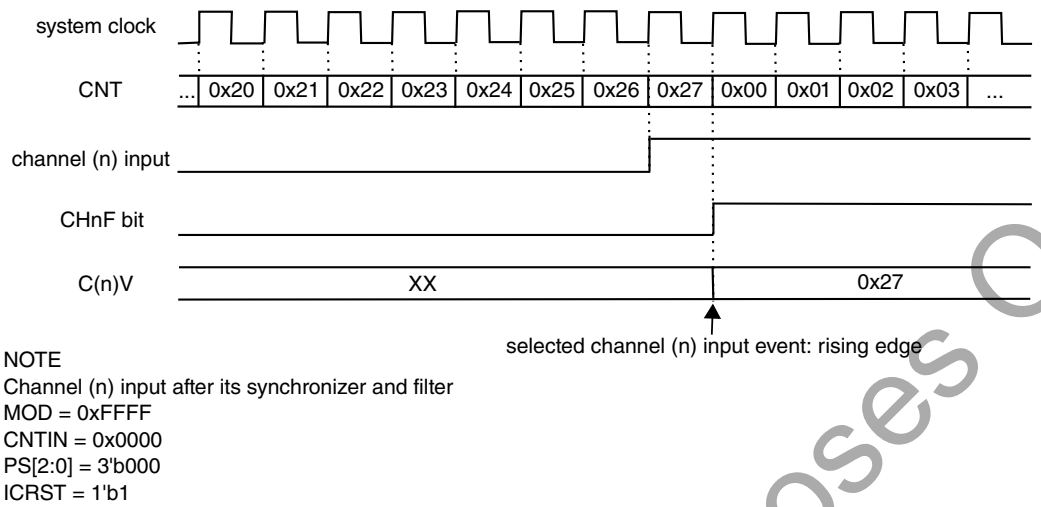


Figure 38-16. Example of the Input Capture mode with ICRST = 1

NOTE

- It is expected that the ICRST bit be set only when the channel is in input capture mode.
- If the FTM counter is reset because the channel is in input capture mode with ICRST = 1, then the prescaler counter (Prescaler) is also reset.

38.5.6 Output Compare mode

The Output Compare mode is selected when:

- DECAPEN = 0
- MCOMBINE = 0
- COMBINE = 0
- CPWMS = 0, and
- MSnB:MSnA = 0:1

In Output Compare mode, the FTM 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.

When a channel is initially configured to Toggle mode, the previous value of the channel output is held until the first output compare event occurs.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1 at the channel (n) match (FTM counter = CnV).

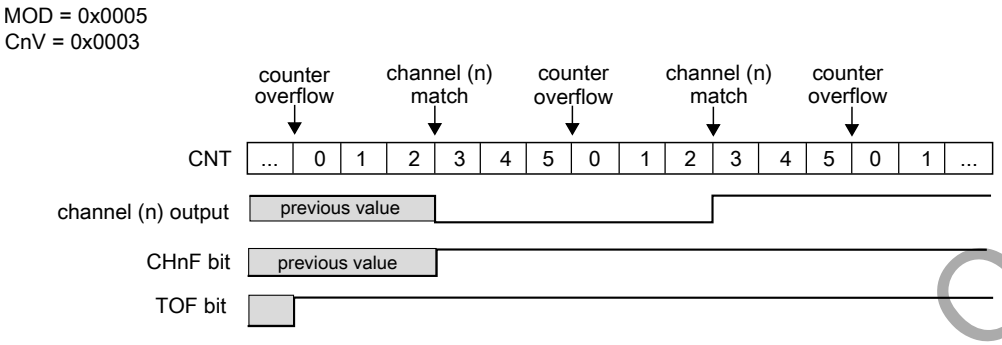


Figure 38-17. Example of the Output Compare mode when the match toggles the channel output

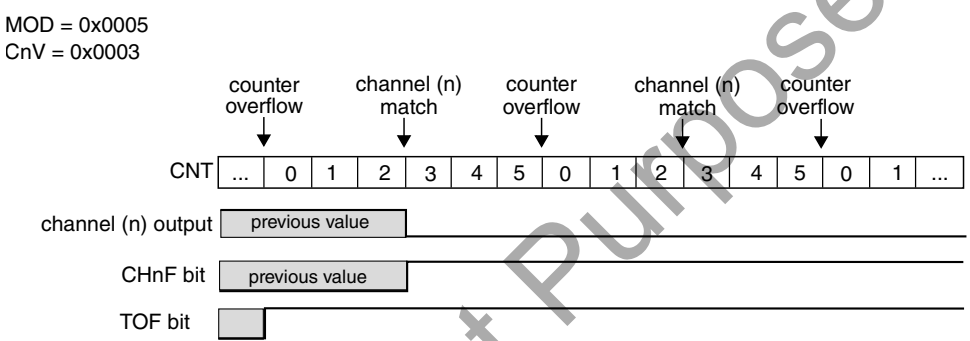


Figure 38-18. Example of the Output Compare mode when the match clears the channel output

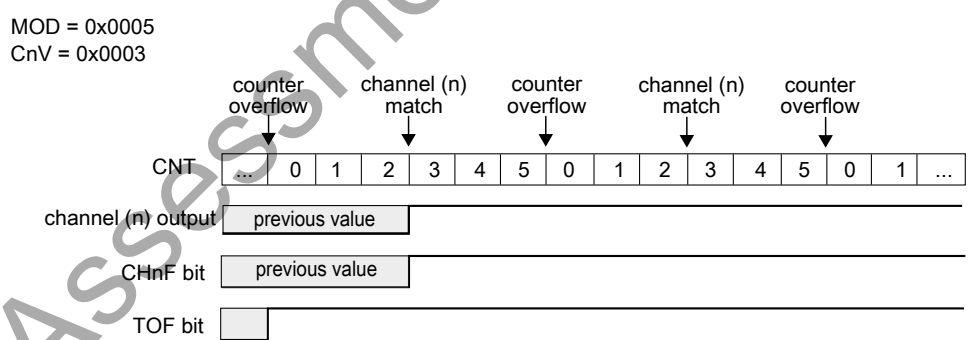


Figure 38-19. Example of the Output Compare mode when the match sets the channel output

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 modified and controlled by FTM.

For Assessment Purposes Only

38.5.7 Edge-Aligned PWM (EPWM) mode

The Edge-Aligned mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- MCOMBINE = 0
- COMBINE = 0
- CPWMS = 0, and
- MSnB = 1

The EPWM period is determined by (MOD − CNTIN + 0x0001) and the pulse width (duty cycle) is determined by (CnV − CNTIN).

The CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1 at the channel (n) match (FTM 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 FTM.

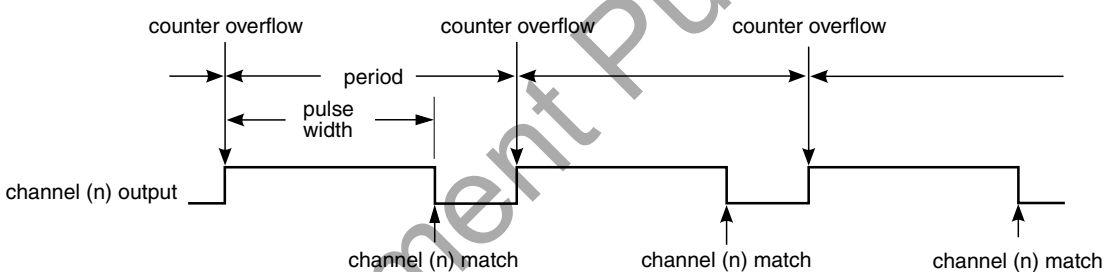


Figure 38-20. EPWM period and pulse width with ELSnB:ELSnA = 1:0

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 FTM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced low at the channel (n) match (FTM counter = CnV). See the following figure.

For Assessment Purposes Only

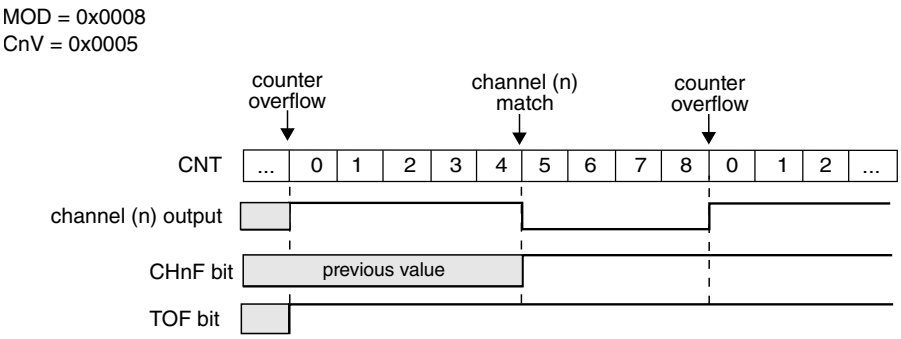


Figure 38-21. EPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced high at the channel (n) match (FTM counter = CnV). See the following figure.

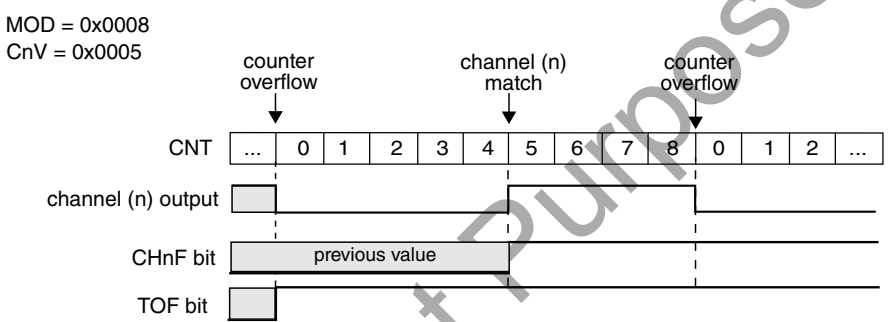


Figure 38-22. EPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000), then the channel (n) output is a 0% duty cycle EPWM signal and CHnF bit is not set even when there is the channel (n) match.

If (CnV > MOD), then the channel (n) output is a 100% duty cycle EPWM signal and CHnF bit is not set. Therefore, MOD must be less than 0xFFFF in order to get a 100% duty cycle EPWM signal.

Note

When CNTIN is different from zero the following EPWM signals can be generated:

- 0% EPWM signal if CnV = CNTIN,
- EPWM signal between 0% and 100% if CNTIN < CnV <= MOD,
- 100% EPWM signal when CNTIN > CnV or CnV > MOD.

For Assessment Purposes Only

38.5.8 Center-Aligned PWM (CPWM) mode

The Center-Aligned mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- MCOMBINE = 0
- COMBINE = 0, and
- CPWMS = 1

The CPWM pulse width (duty cycle) is determined by 2 × (CnV − CNTIN) and the period is determined by 2 × (MOD − CNTIN). 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 FTM counter counts up until it reaches MOD and then counts down until it reaches CNTIN.

The CHnF bit is set and channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (FTM counter = CnV) when the FTM counting is down (at the begin of the pulse width) and when the FTM 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 aligned with the value of CNTIN.

The other channel modes are not compatible with the up-down counter (CPWMS = 1). Therefore, all FTM channels must be used in CPWM mode when (CPWMS = 1).

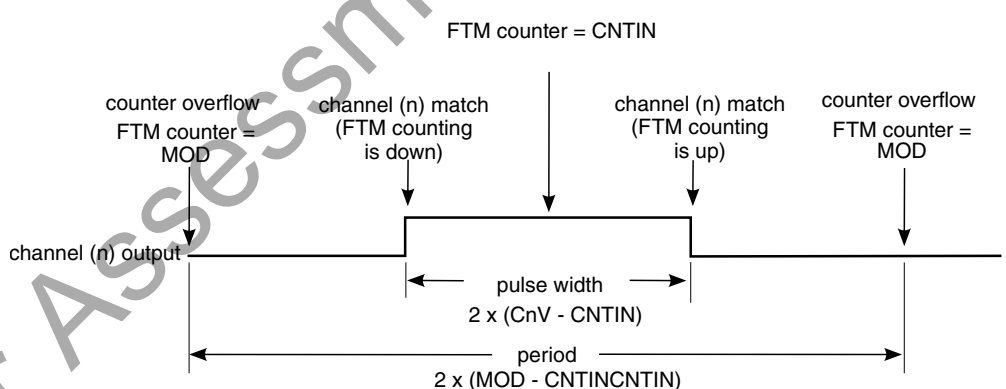


Figure 38-23. CPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the FTM 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 FTM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the channel (n) match (FTM counter = CnV) when counting down, and it is forced low at the channel (n) match when counting up. See the following figure.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

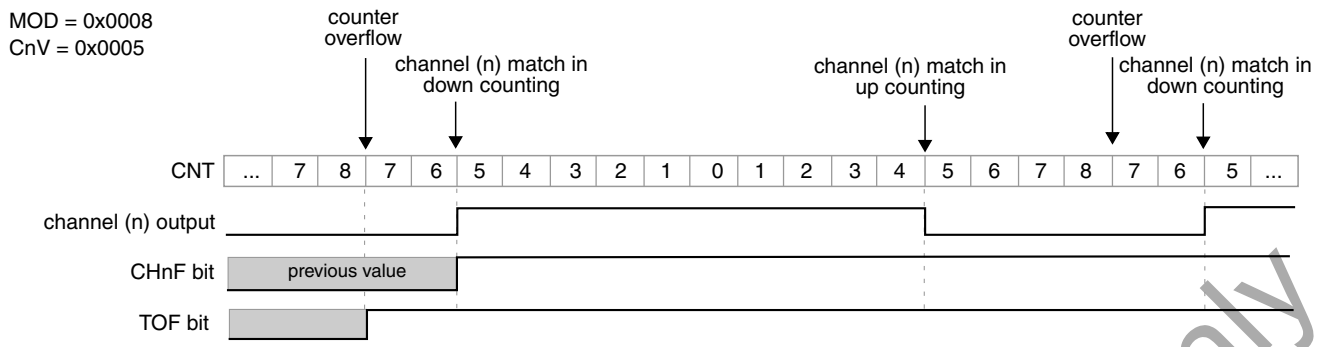


Figure 38-24. CPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the channel (n) match (FTM counter = CnV) when counting down, and it is forced high at the channel (n) match when counting up. See the following figure.

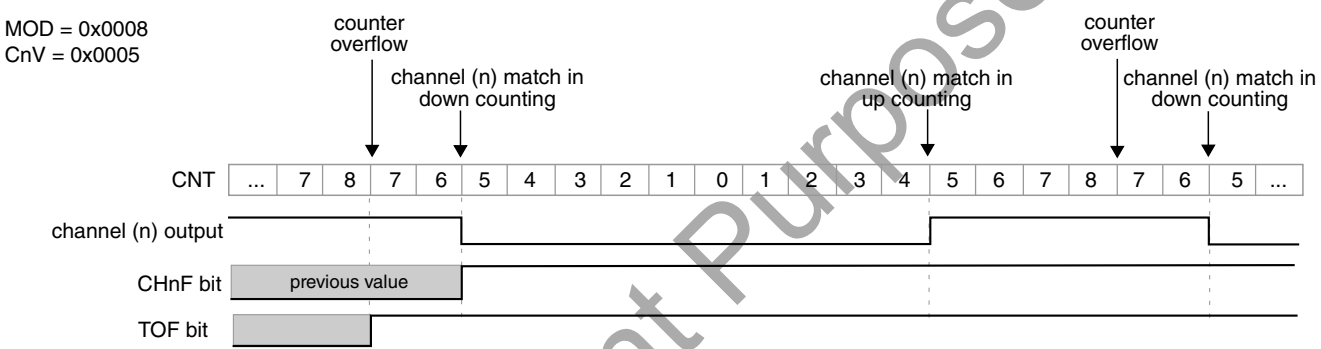


Figure 38-25. CPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000) or CnV is a negative value, that is (CnV[15] = 1), then the channel (n) output is a 0% duty cycle CPWM signal and CHnF bit is not set even when there is the channel (n) match.

If CnV is a positive value, that is (CnV[15] = 0), (CnV ≥ MOD), and (MOD ≠ 0x0000), then the channel (n) output is a 100% duty cycle CPWM signal and CHnF bit is not set even when there is the channel (n) match. This implies that the usable range of periods set by MOD is 0x0001 through 0x7FFE, 0x7FFF if you do not need to generate a 100% duty cycle CPWM signal. This is not a significant limitation because the resulting period is much longer than required for normal applications.

The CPWM mode must not be used when the FTM counter is a free running counter.

38.5.9 Combine mode

The Combine mode is selected when:

- QUADEN = 0

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- DECAPEN = 0
- MCOMBINE = 0
- COMBINE = 1, and
- CPWMS = 0

In Combine 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 − CNTIN + 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 (FTM 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 (FTM counter = C(n+1)V).

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n +1)V). It is forced high at the channel (n) match (FTM counter = C(n)V). See the following figure.

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced high at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n +1)V). It is forced low at the channel (n) match (FTM counter = C(n)V). See the following figure.

In Combine mode, the ELS(n+1)B and ELS(n+1)A bits are not used in the generation of the channels (n) and (n+1) output. However, if (ELSnB:ELSnA = 0:0) then the channel (n) output is not controlled by FTM, and if (ELS(n+1)B:ELS(n+1)A = 0:0) then the channel (n+1) output is not controlled by FTM.

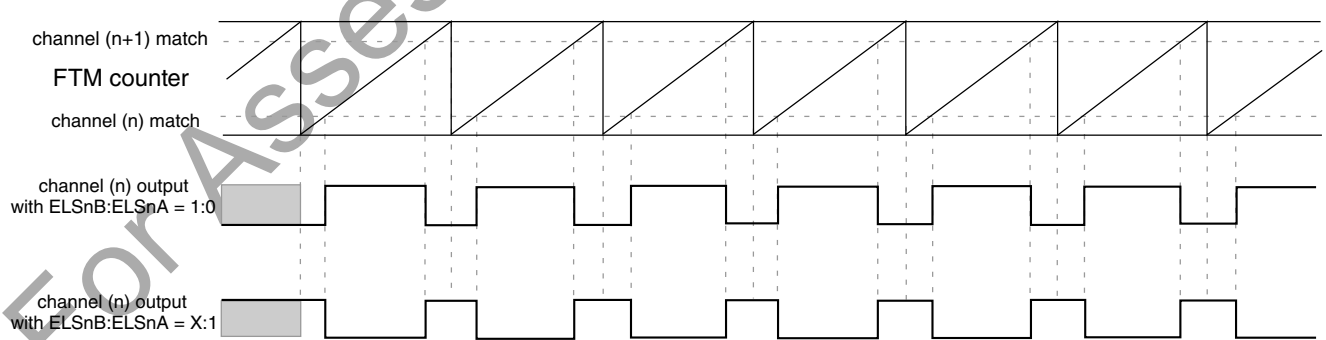


Figure 38-26. Combine mode

The following figures illustrate the PWM signals generation using Combine mode.

For Assessment Purposes Only

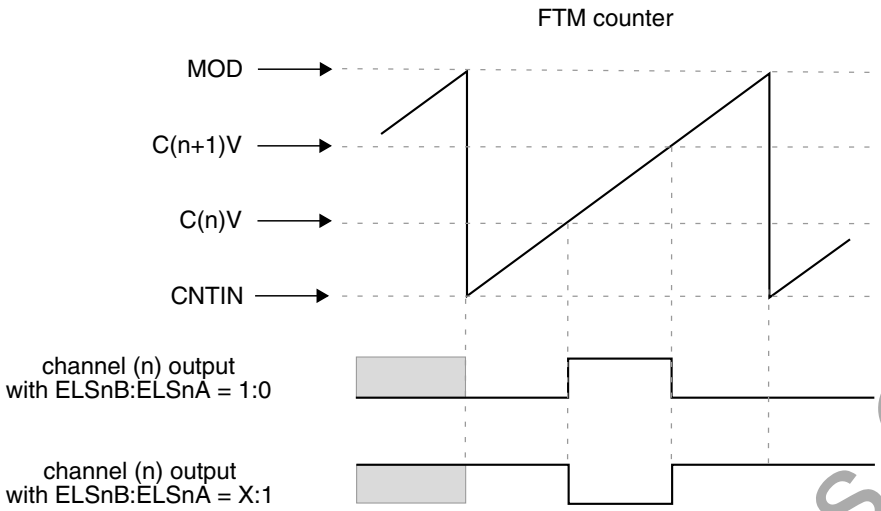


Figure 38-27. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V < C(n+1)V)

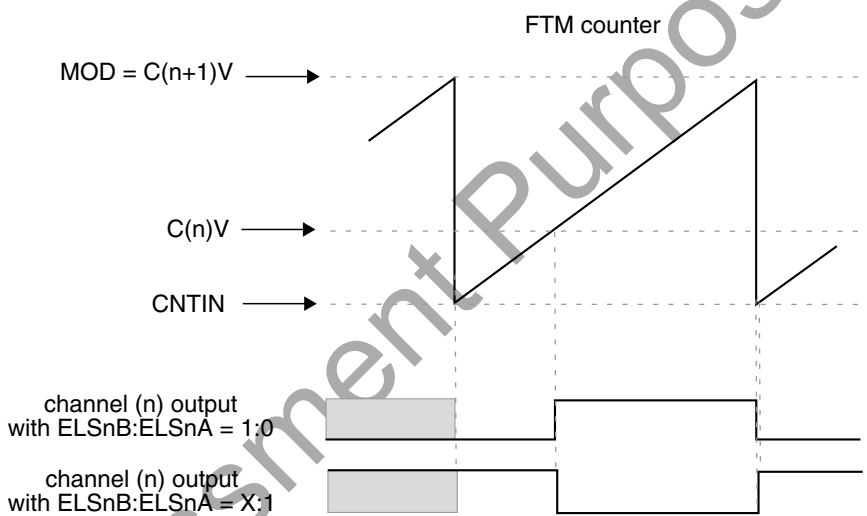


Figure 38-28. Channel (n) output if (CNTIN < C(n)V < MOD) and (C(n+1)V = MOD)

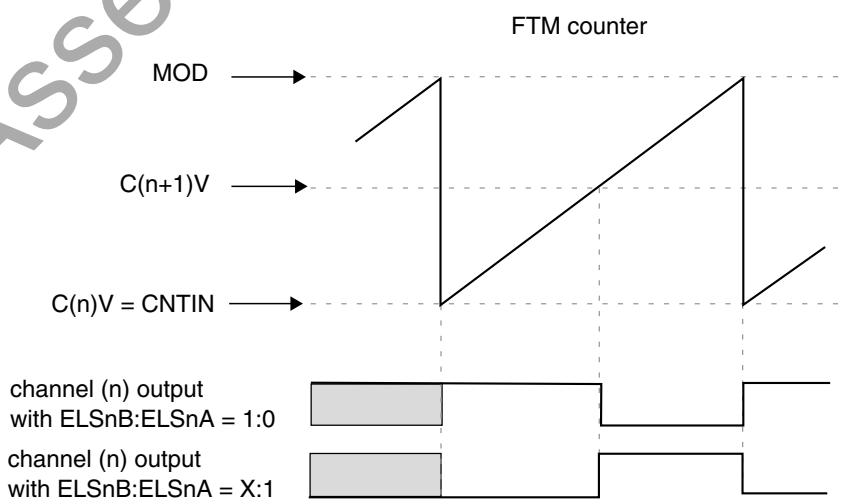


Figure 38-29. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD)

For Assessment Purposes Only

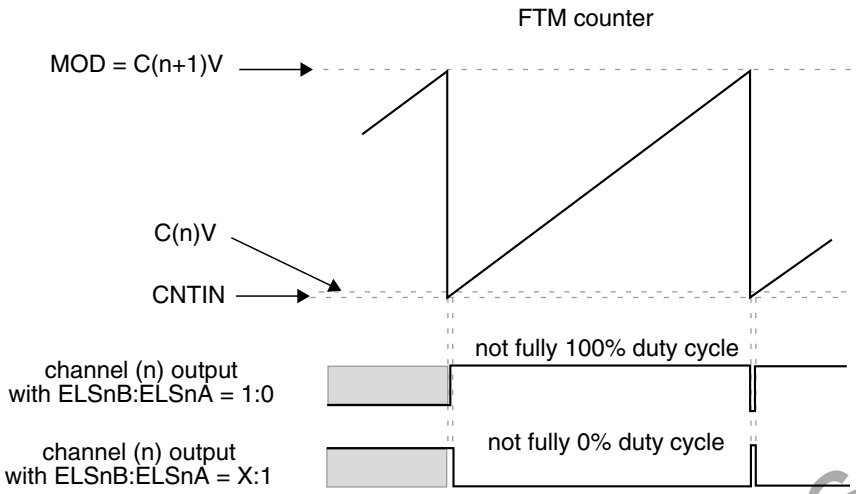


Figure 38-30. Channel (n) output if (CNTIN < C(n)V < MOD) and (C(n)V is Almost Equal to CNTIN) and (C(n+1)V = MOD)

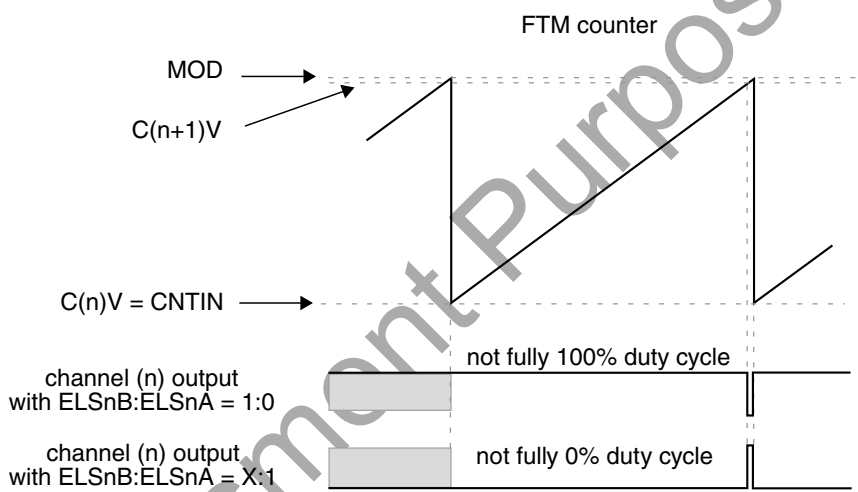


Figure 38-31. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD) and (C(n+1)V is Almost Equal to MOD)

For Assessment Purposes Only

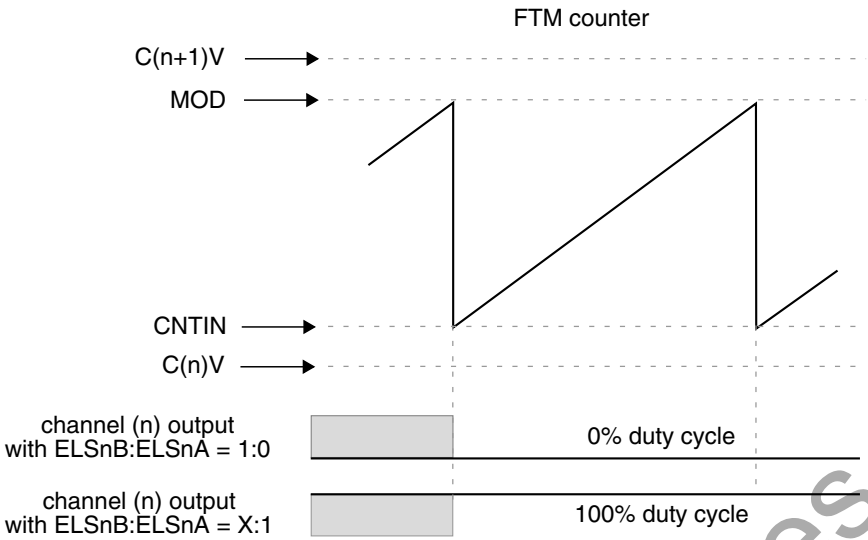


Figure 38-32. Channel (n) output if C(n)V and C(n+1)V are not between CNTIN and MOD

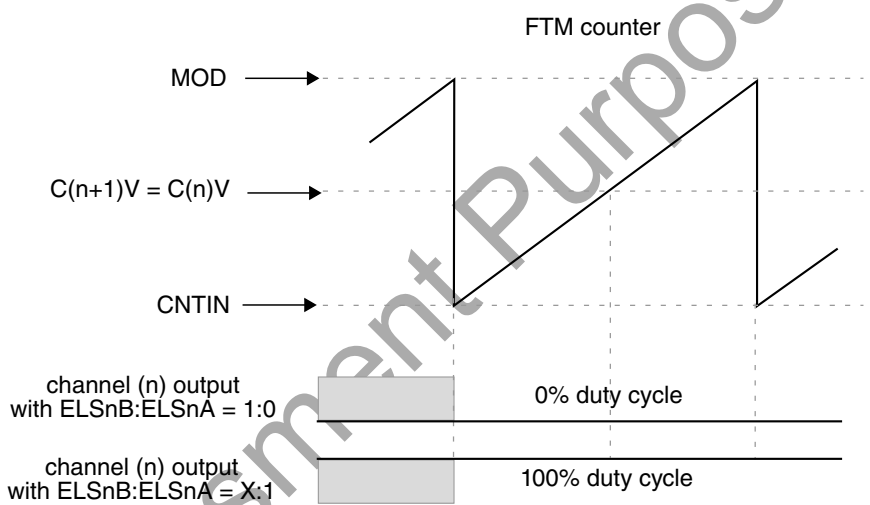


Figure 38-33. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V = C(n+1)V)

For Assessment Purposes Only

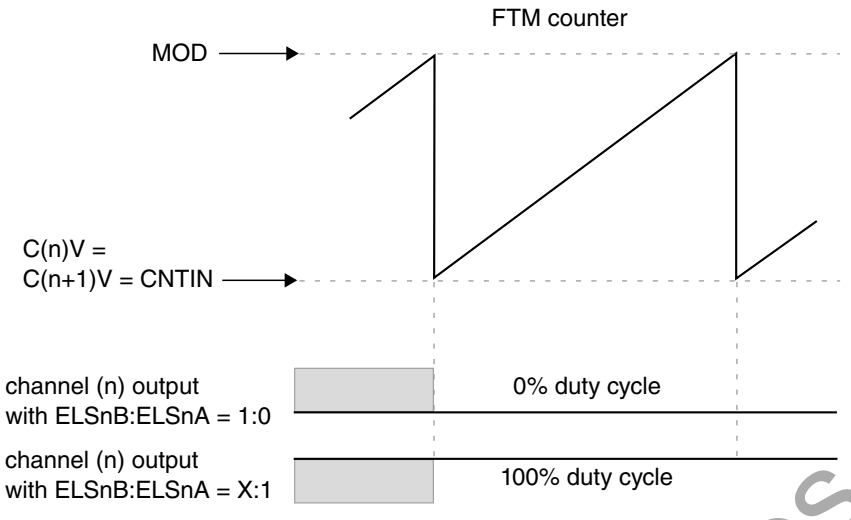


Figure 38-34. Channel (n) output if (C(n)V = C(n+1)V = CNTIN)

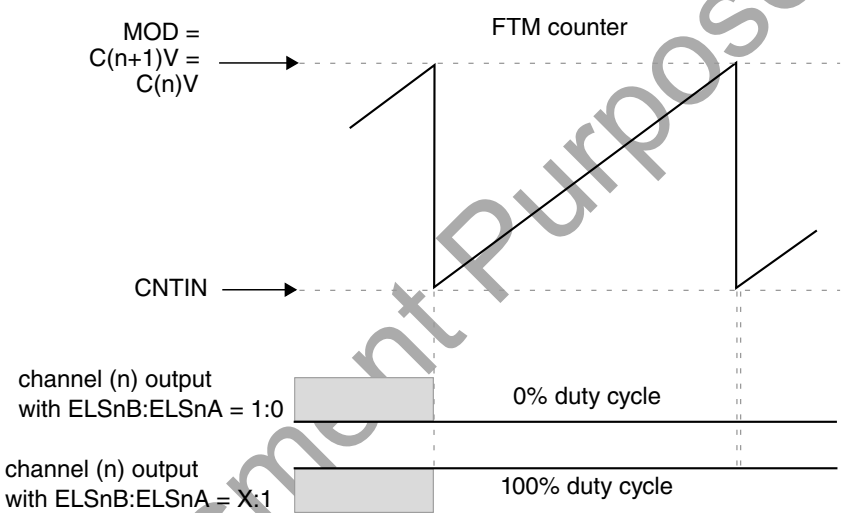


Figure 38-35. Channel (n) output if (C(n)V = C(n+1)V = MOD)

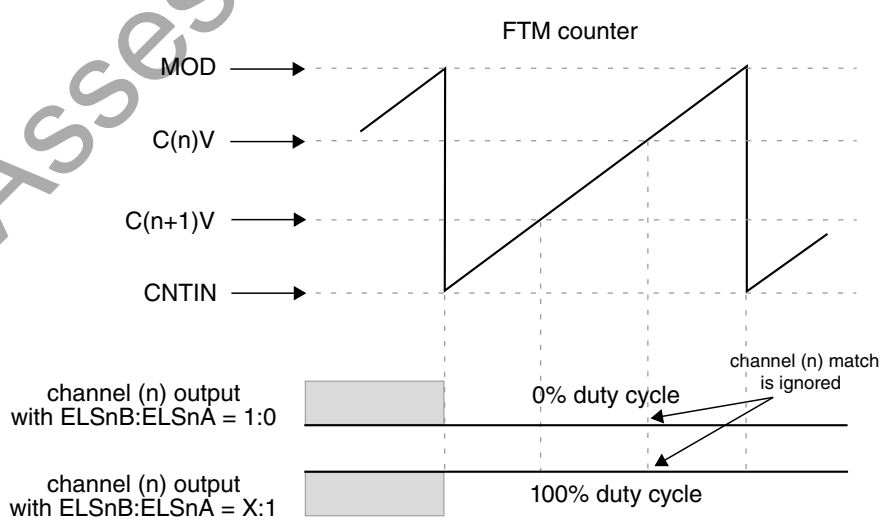


Figure 38-36. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V > C(n+1)V)

For Assessment Purposes Only

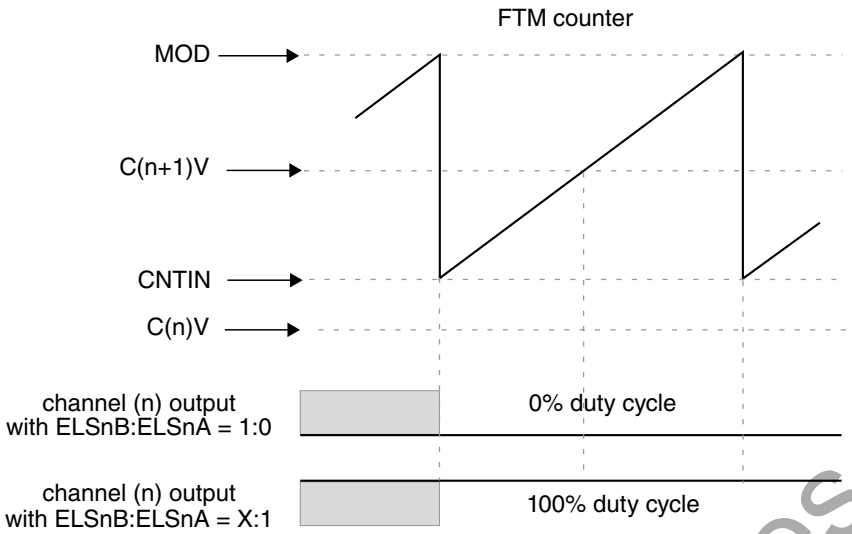


Figure 38-37. Channel (n) output if (C(n)V < CNTIN) and (CNTIN < C(n+1)V < MOD)

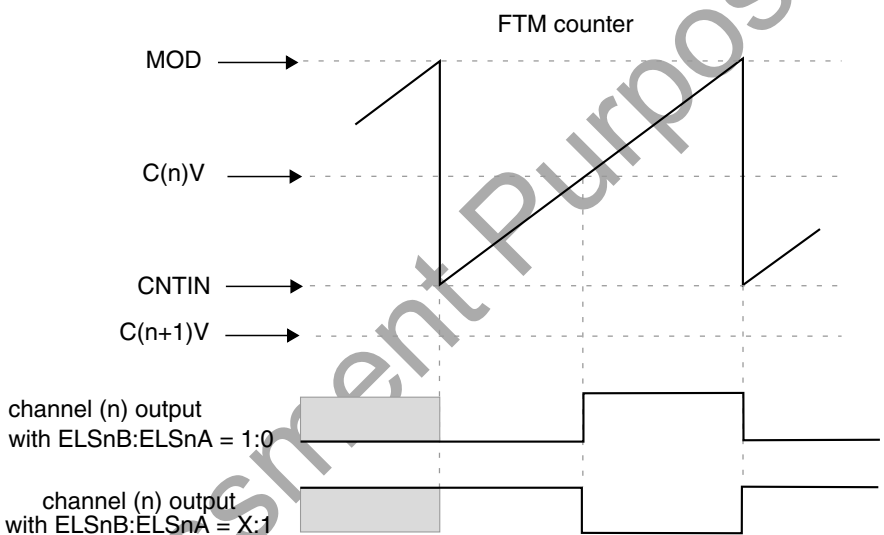


Figure 38-38. Channel (n) output if (C(n+1)V < CNTIN) and (CNTIN < C(n)V < MOD)

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

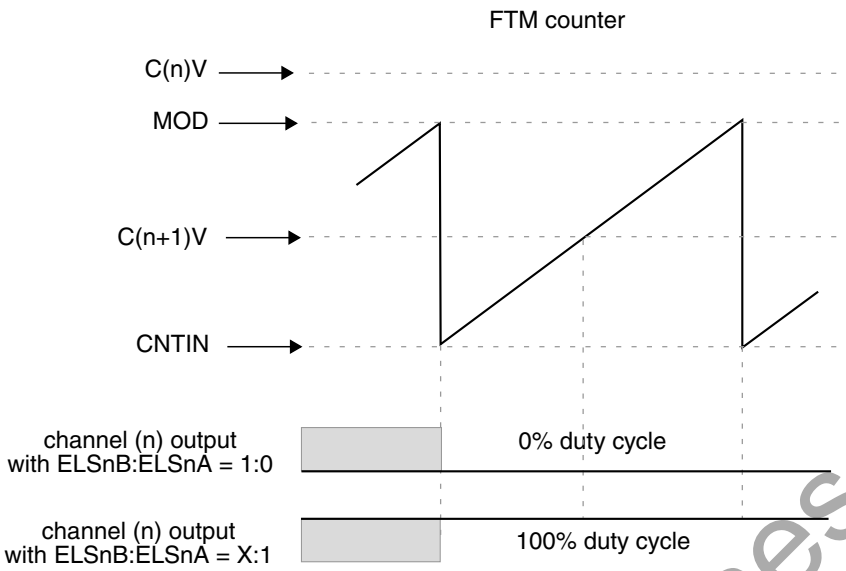


Figure 38-39. Channel (n) output if (C(n)V > MOD) and (CNTIN < C(n+1)V < MOD)

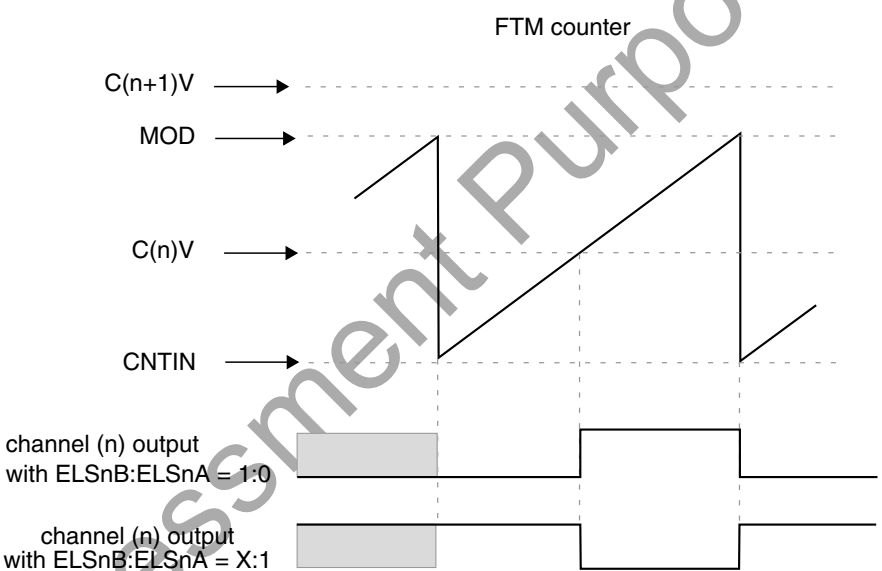


Figure 38-40. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V < MOD)

For Assessment Purposes Only

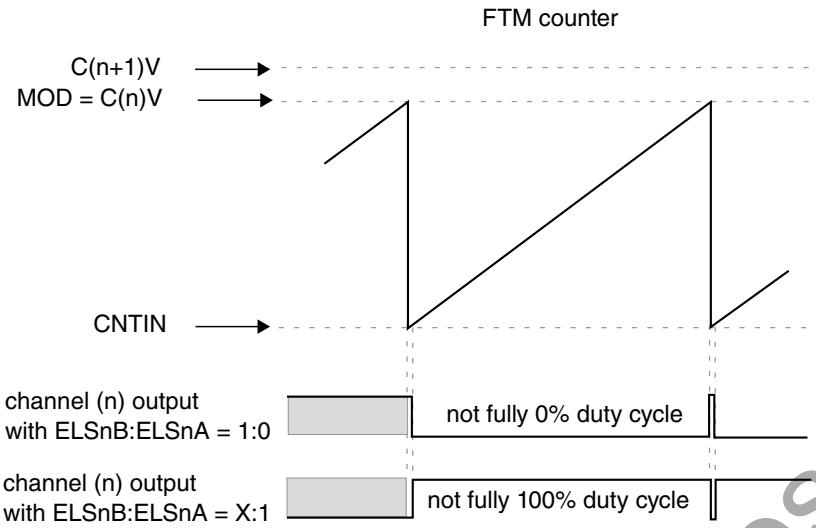


Figure 38-41. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V = MOD)

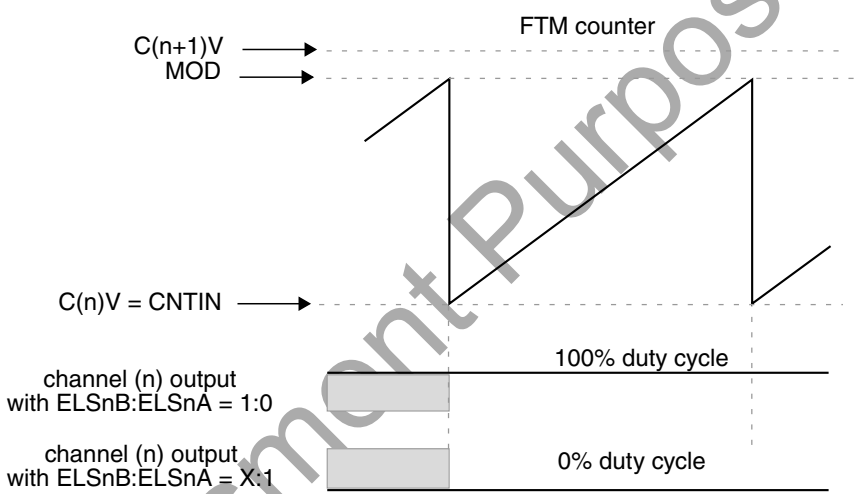


Figure 38-42. Channel (n) output if (C(n)V = CNTIN) and (C(n+1)V > MOD)

38.5.9.1 Asymmetrical PWM

In Combine mode and Modified Combine PWM Mode, the PWM first edge (channel (n) match: FTM counter = C(n)V) is independent of the PWM second edge (channel (n+1) match: FTM counter = C(n+1)V).

38.5.10 Modified Combine PWM Mode

The Modified Combine PWM mode is selected when:

- QUADEN = 0
- DECAPEN = 0

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- MCOMBINE = 1
- COMBINE = 1, and
- CPWMS = 0

The Modified Combine PWM mode is intended to support the generation of PWM signals where the period is not modified while the signal is being generated, but the duty cycle will be varied. In this mode, an even channel (n) and adjacent odd channel (n+1) are combined to generate a PWM signal in the channel (n) output. Thus, the channel (n) match edge is fixed and the channel (n+1) match edge can be varied.

The Modified Combine PWM mode should be used with more than one pair of channels.

In the Modified Combine PWM mode, assuming that CNTIN ≥ 0, MOD > 0, and CNTIN < MOD:

- The PWM period is determined by (MOD - CNTIN + 0x0001);
- The channel (n) PWM duty cycle is calculated according to the following table.

Table 38-7. Modified Combine PWM Mode - Duty Cycles

Channel (n) PWM Duty Cycle	Condition
0% duty cycle	For CNTIN ≤ (C(n)V and C(n+1)V) ≤ MOD: C(n)V = C(n+1)V
duty cyle between 0% and 100%	For CNTIN ≤ (C(n)V and C(n+1)V) ≤ MOD: • if (C(n)V < C(n+1)V), then the duty cycle is (C(n+1)V - C(n)V) • if (C(n)V > C(n+1)V), then the duty cycle is [(MOD - C(n)V) + (C(n+1)V - CNTIN)]
100% duty cyle	CNTIN ≤ C(n)V ≤ MOD and C(n+1)V > MOD

The CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (FTM 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 (FTM counter = C(n+1)V).

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the channel (n) match (FTM counter = C(n)V) and it is forced low at the channel (n+1) match (FTM counter = C(n+1)V).

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the channel (n) match (FTM counter = C(n)V) and it is forced high at the channel (n+1) match (FTM counter = C(n+1)V).

In Modified Combine PWM mode, the ELS(n+1)B and ELS(n+1)A bits are not used in the generation of the channels (n) and (n+1) output. However, if (ELSnB:ELSnA = 0:0) then the channel (n) output is not controlled by FTM, and if (ELS(n+1)B:ELS(n+1)A = 0:0) then the channel (n+1) output is not controlled by FTM.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

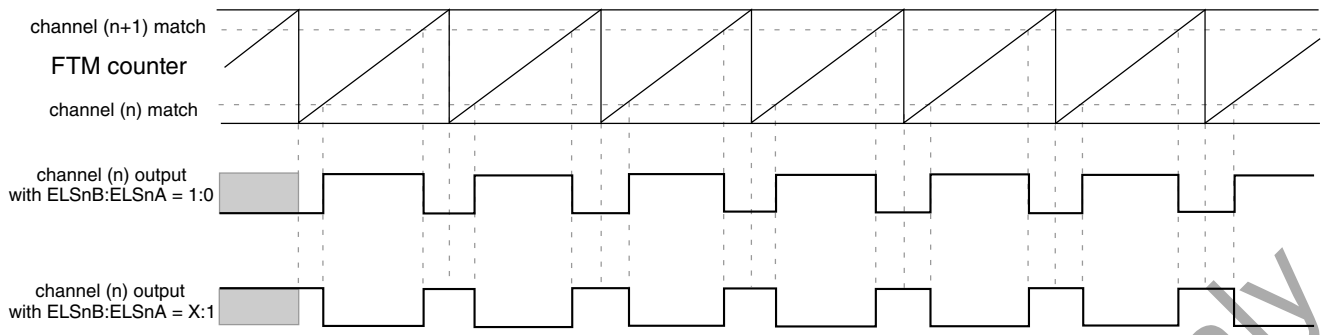


Figure 38-43. Modified Combine PWM Mode

The Modified Combine PWM mode allows the offset addition of the duty cyle, thus, in some cases, the C(n+1)V match can happen on the next FTM counter period. For CNTIN ≥ 0, MOD > 0, and CNTIN < MOD, this situation happens when C(n)V > C(n+1)V.

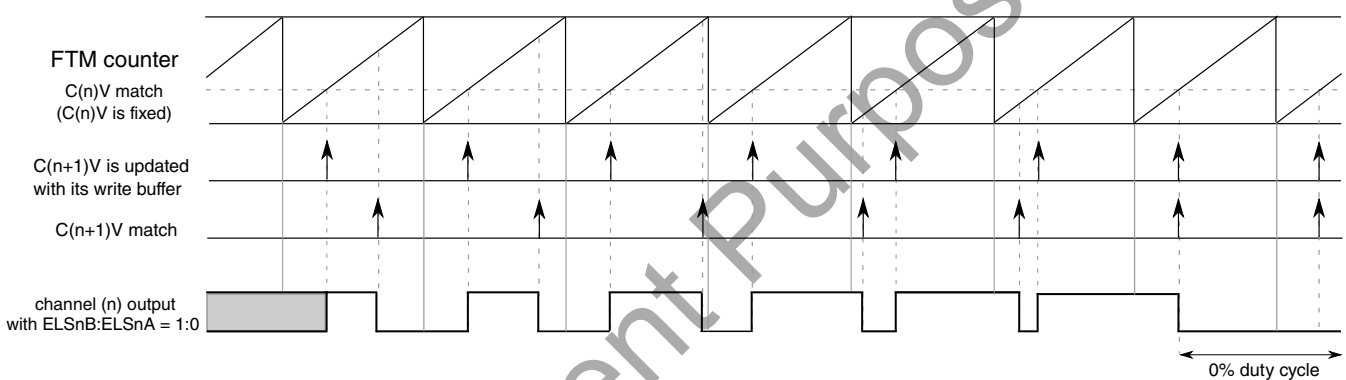


Figure 38-44. Modified Combine PWM Mode Examples

If more than one pair of channels are configured in Modified Combine PWM Mode, it is possible to fix an offset for the channel (n) match edge of each pair with respect to other pairs. This behavior is useful in the generation of lighting PWM control signals where it is desirable that edges are not coincident with each other pair to help eliminate noise generation. The C(n)V register value is the shift of the PWM pulse with respect to the beginning of FTM counter period (FTM counter = CNTIN).

For Assessment Purposes Only

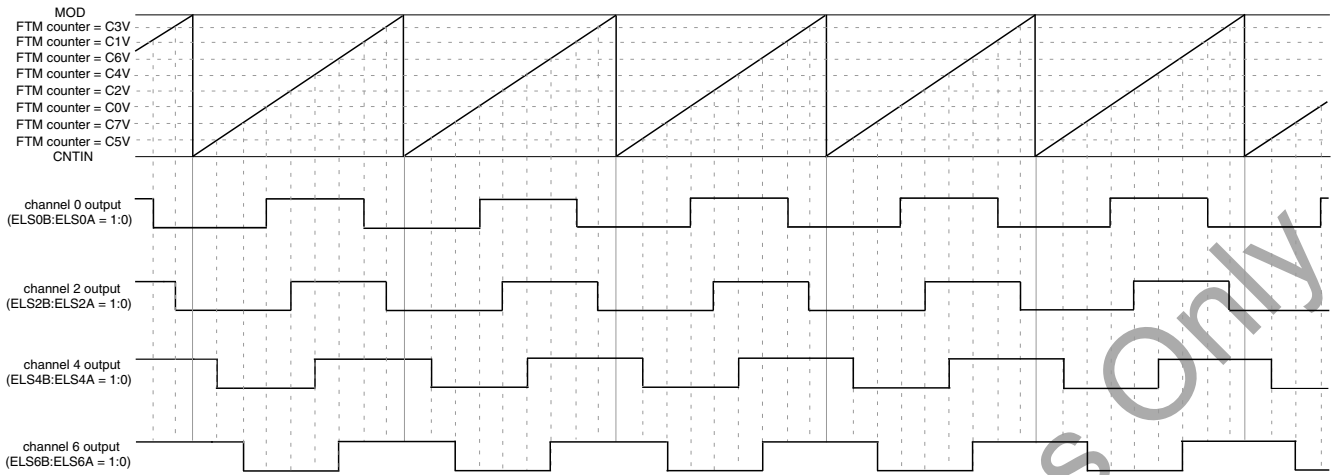


Figure 38-45. Example of Four Pairs of Channels in Modified Combine PWM Mode

NOTE

The Modified Combine PWM Mode can be used with Complementary Mode, Software Output Control Mode, Polarity Control, and External Trigger.

38.5.10.1 Synchronization

In the Modified Combine Mode, the FTM counter period (CNTIN and/or MOD) and the channel (n) match value (C(n)V) should be configured only when the FTM clock is disabled (CLKS[1:0] = 0:0). The initial value of the channel (n+1) match (C(n+1)V) can be configured when the FTM clock is disabled (CLKS[1:0] = 0:0). See Registers updated from write buffers.

To allow that the duty cycle can be varied in the Modified Combine Mode, the C(n+1)V register is updated with the new value of its write buffer on the channel (n) match (FTM counter = C(n)V).

NOTE

In the Modified Combine Mode, the SYNCEN(n) bit for the channels (n) and (n+1) should be zero. So, the following features are not available for this mode.

- C(n)V and C(n+1)V register synchronization;
- Half and full cycle reload;
- Global Load.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.5.11 Complementary Mode

The Complementary mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMP = 1

In Complementary mode, the channel (n+1) output is the inverse of the channel (n) output.

So, the channel (n+1) output is the same as the channel (n) output when:

- QUADEN = 0
- DECAPEN = 0
- COMP = 0

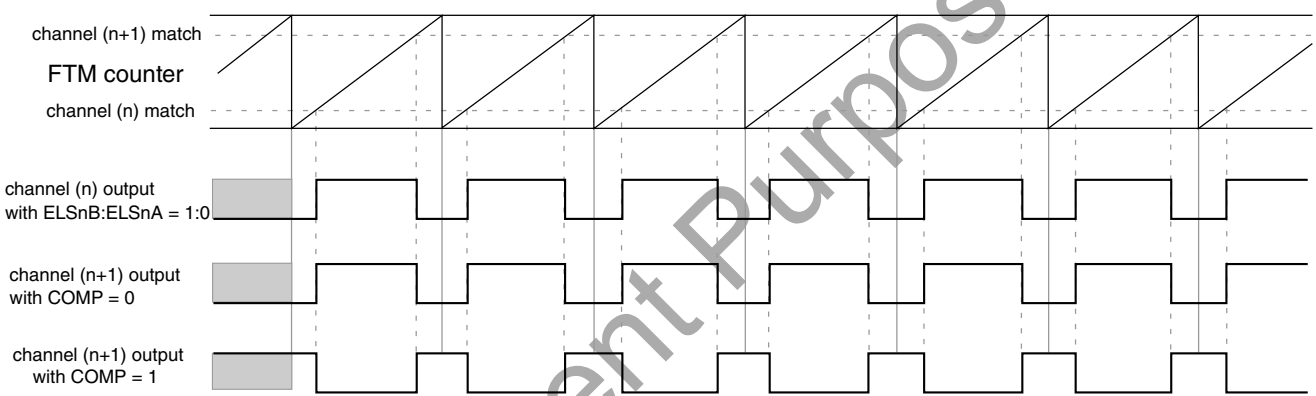


Figure 38-46. Channel (n+1) output in Complementary mode with (ELSnB:ELSnA = 1:0)

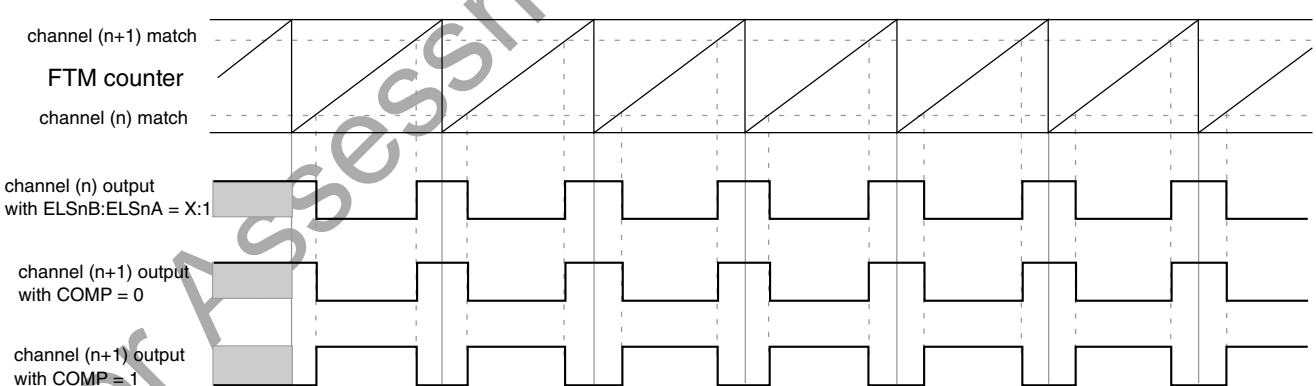


Figure 38-47. Channel (n+1) output in Complementary mode with (ELSnB:ELSnA = X:1)

NOTE

The Complementary Mode is not available on Output Compare mode.

For Assessment Purposes Only

38.5.12 Registers updated from write buffers

FTM has many ways to synchronize PWM registers. Current implementation allows to bypass the buffers, use legacy and PWM synchronization (hardware and software trigger) and it is also possible to use a half or full cycle reload strategy.

38.5.12.1 CNTIN register update

The following table describes when CNTIN register is updated:

Table 38-8. CNTIN register update

When	Then CNTIN register is updated
CLKS[1:0] = 0:0	When CNTIN register is written, independent of FTMEN bit.
• FTMEN = 0, or • CNTINC = 0	At the next system clock after CNTIN was written.
• FTMEN = 1, • SYNCMODE = 1, and • CNTINC = 1	By the CNTIN register synchronization.
• CNTINC = 1, and • LDOK = 1	By the Half and full cycle reload.

38.5.12.2 MOD and HCR registers update

The following table describes when MOD or HCR registers are updated:

Table 38-9. MOD and HCR updates

When	Then MOD or HCR is updated
CLKS[1:0] = 0:0	When MOD (or HCR) is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the CPWMS bit, that is: • If the selected mode is not CPWM then MOD (or HCR) is updated after MOD (or HCR) register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM then MOD (or HCR) register is updated after MOD (or HCR) register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	By the MOD register synchronization. HCR follows the same procedure of MOD register in this case.
• LDOK = 1	By the Half and full cycle reload.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.5.12.3 CnV register update

The following table describes when CnV register is updated:

Table 38-10. CnV register update

When	Then CnV register is updated
CLKS[1:0] = 0:0	When CnV register is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the selected mode, that is: • If the selected mode is Output Compare, then CnV register is updated on the next FTM counter change, 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 FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	According to the selected mode, that is: • If the selected mode is output compare then CnV register is updated according to the SYNCEN bit. If (SYNCEN = 0) then CnV register is updated after CnV register was written at the next change of the FTM counter, the end of the prescaler counting. If (SYNCEN = 1) then CnV register is updated by the C(n)V and C(n+1)V register synchronization. • If the selected mode is not output compare and (SYNCEN = 1) then CnV register is updated by the C(n)V and C(n+1)V register synchronization.
• SYNCEN = 1, and • LDOK = 1	By the Half and full cycle reload.

38.5.13 PWM synchronization

The PWM synchronization provides an opportunity to update the MOD, HCR, CNTIN, CnV, OUTMASK, INVCTRL and SWOCTRL registers with their buffered value and force the FTM counter to the CNTIN register value.

Note

The legacy PWM synchronization (SYNCMODE = 0) is a subset of the enhanced PWM synchronization (SYNCMODE = 1). Thus, only the enhanced PWM synchronization must be used.

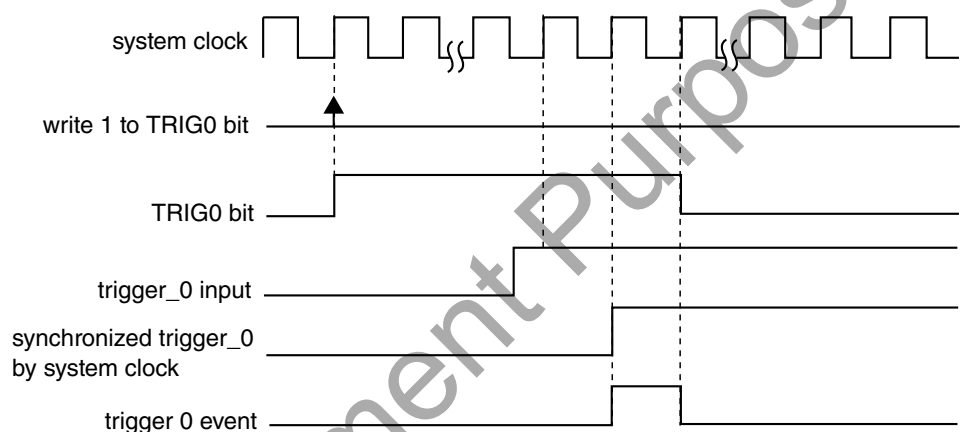
For Assessment

38.5.13.1 Hardware trigger

Three hardware trigger signal inputs of the FTM module are enabled when TRIGn = 1, where n = 0, 1 or 2 corresponding to each one of the input signals, respectively. The hardware trigger input n is synchronized by the system clock. The PWM synchronization with hardware trigger is initiated when a rising edge is detected at the enabled hardware trigger inputs.

If (HWTRIGMODE = 0) then the TRIGn bit is cleared when 0 is written to it or when the trigger n event is detected.

In this case, if two or more hardware triggers are enabled (for example, TRIG0 and TRIG1 = 1) and only trigger 1 event occurs, then only TRIG1 bit is cleared. If a trigger n event occurs together with a write setting TRIGn bit, then the synchronization is initiated, but TRIGn bit remains set due to the write operation.



Note
All hardware trigger inputs have the same behavior.

Figure 38-48. Hardware trigger event with HWTRIGMODE = 0

If HWTRIGMODE = 1, then the TRIGn bit is only cleared when 0 is written to it.

NOTE

The HWTRIGMODE bit must be 1 only with enhanced PWM synchronization (SYNCMODE = 1).

38.5.13.2 Software trigger

A software trigger event occurs when 1 is written to the SYNC[SWSYNC] bit. The SWSYNC bit is cleared when 0 is written to it or when the PWM synchronization, initiated by the software event, is completed.

For Assessment Purposes Only

If another software trigger event occurs (by writing another 1 to the SWSYNC bit) at the same time the PWM synchronization initiated by the previous software trigger event is ending, a new PWM synchronization is started and the SWSYNC bit remains equal to 1.

If SYNCMODE = 0 then the SWSYNC bit is also cleared by FTM according to PWMSYNC and REINIT bits. In this case if (PWMSYNC = 1) or (PWMSYNC = 0 and REINIT = 0) then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see Boundary cycle and loading points and the following figure. If (PWMSYNC = 0) and (REINIT = 1) then SWSYNC bit is cleared when the software trigger event occurs.

If SYNCMODE = 1 then the SWSYNC bit is also cleared by FTM according to the SWRSTCNT bit. If SWRSTCNT = 0 then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see the following figure. If SWRSTCNT = 1 then SWSYNC bit is cleared when the software trigger event occurs.

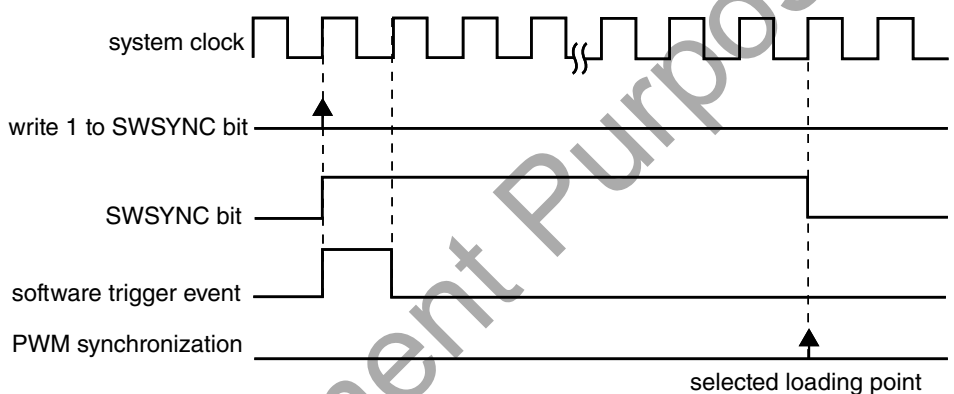


Figure 38-49. Software trigger event

38.5.13.3 Boundary cycle and loading points

The boundary cycle definition is important for the loading points for the registers MOD, CNTIN, and C(n)V.

In Up counting mode, the boundary cycle is defined as when the counter wraps to its initial value (CNTIN). If in Up-down counting mode, then the boundary cycle is defined as when the counter turns from down to up counting and when from up to down counting.

The following figure shows the boundary cycles and the loading points for the registers. In the Up Counting mode, the loading points are enabled if one of CNTMIN or CTMAX bits are 1. In the Up-Down Counting mode, the loading points are selected by CNTMIN and CNTMAX bits, as indicated in the figure. These loading points are safe places for register updates thus allowing a smooth transitions in PWM waveform generation.

For Assessment Purposes Only

For both counting modes, if neither CNTMIN nor CNTMAX are 1, then the boundary cycles are not used as loading points for registers updates. See the register synchronization descriptions in the following sections for details.

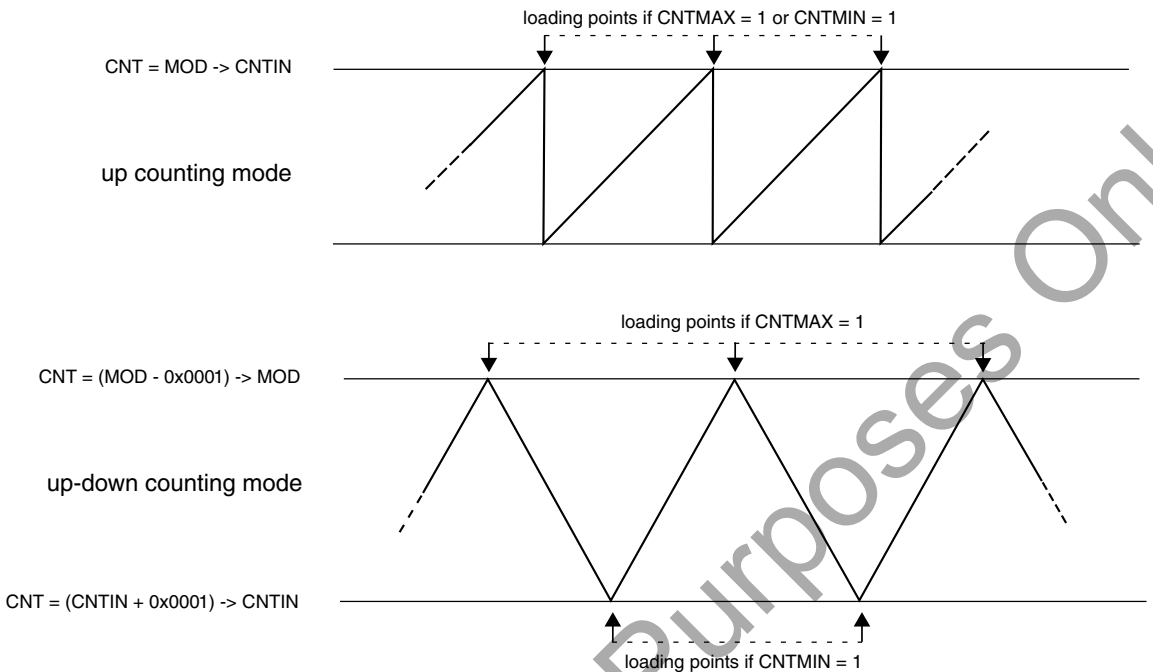


Figure 38-50. Boundary cycles and loading points

38.5.13.4 MOD register synchronization

The MOD register synchronization updates the MOD register with its buffer value. This synchronization is enabled if (FTMEN = 1).

The MOD register synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, it is expected that the MOD register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the MOD register synchronization depends on SWWRBUF, SWRSTCNT, HWWRBUF, and HWRSTCNT bits according to this flowchart:

For Assessment Purposes Only

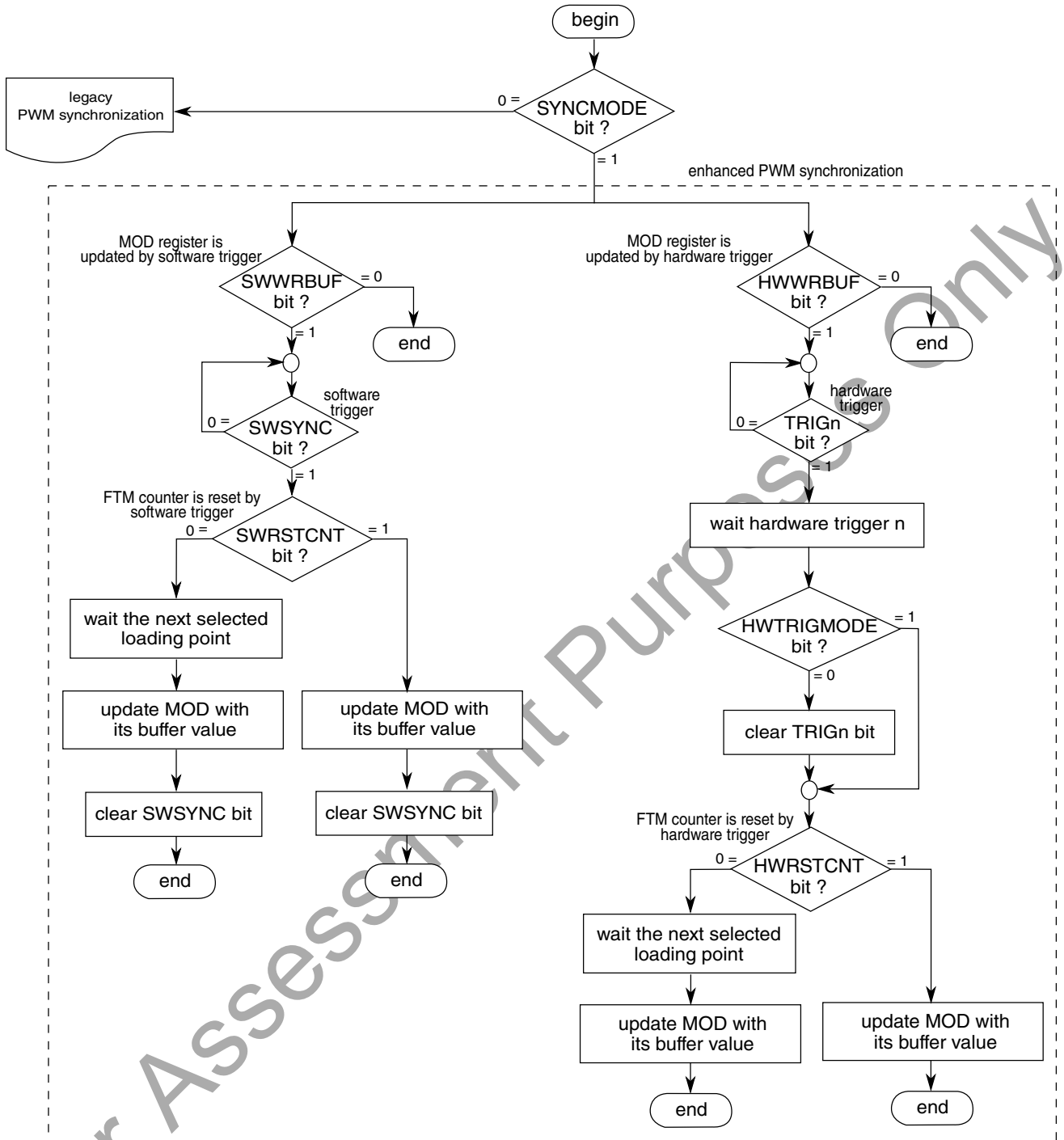


Figure 38-51. MOD register synchronization flowchart

For Assessment Purposes Only

In the case of legacy PWM synchronization, the MOD register synchronization depends on PWMSYNC and REINIT bits according to the following description.

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 0), then this synchronization is made on the next selected loading point after an enabled trigger event takes place. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected

loading point. If the trigger event was a hardware trigger, then the trigger enable bit (TRIGn) is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

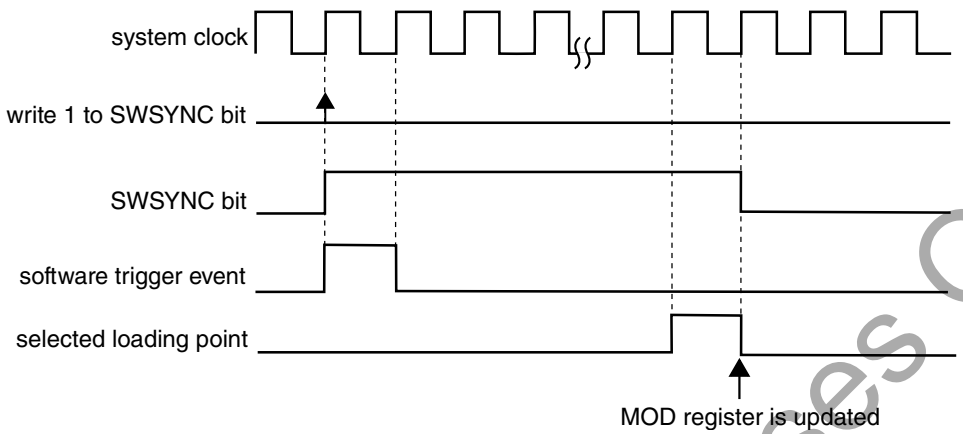


Figure 38-52. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 0), and software trigger was used

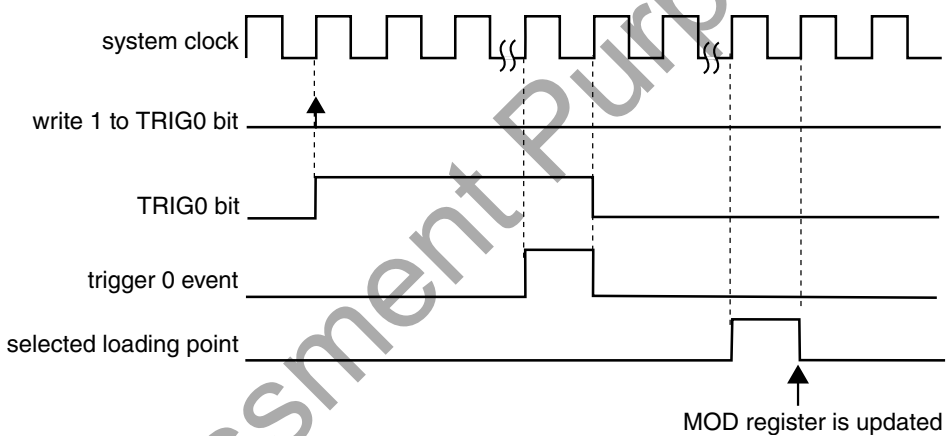


Figure 38-53. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 0), and a hardware trigger was used

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 1), then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

For Assessment Purposes Only

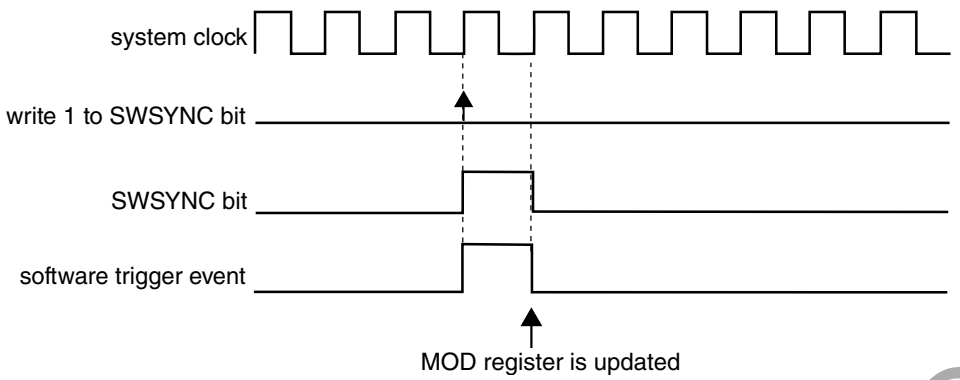


Figure 38-54. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 1), and software trigger was used

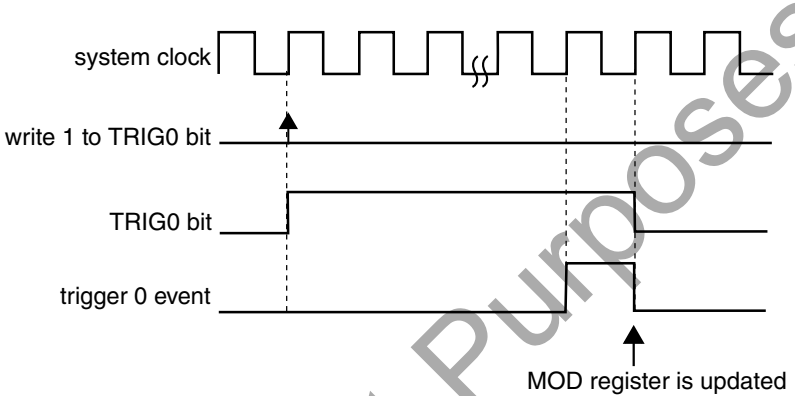


Figure 38-55. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 1), and a hardware trigger was used

If (SYNCMODE = 0) and (PWMSYNC = 1), then this synchronization is made on the next selected loading point after the software trigger event takes place. The SWSYNC bit is cleared on the next selected loading point:

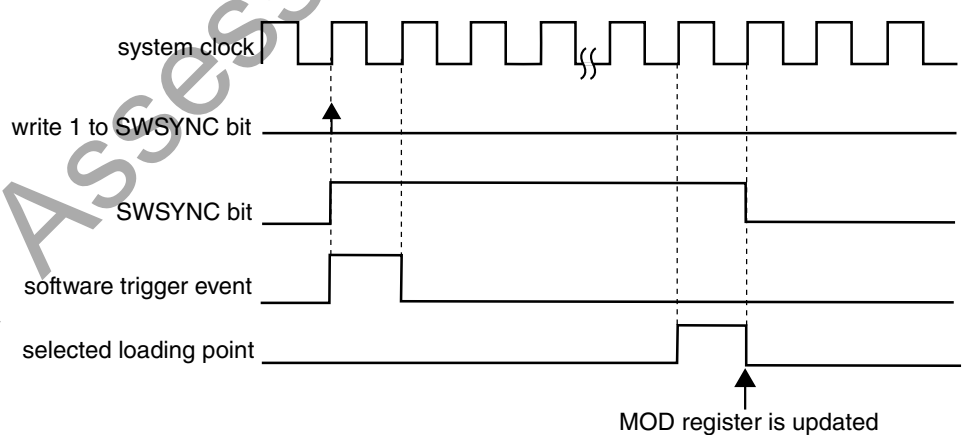


Figure 38-56. MOD synchronization with (SYNCMODE = 0) and (PWMSYNC = 1)

For Assessment Purposes Only

38.5.13.5 CNTIN register synchronization

The CNTIN register synchronization updates the CNTIN register with its buffer value.

This synchronization is enabled if (FTMEN = 1), (SYNCMODE = 1), and (CNTINC = 1). The CNTIN register synchronization can be done only by the enhanced PWM synchronization (SYNCMODE = 1). The synchronization mechanism is the same as the MOD register synchronization done by the enhanced PWM synchronization; see MOD register synchronization.

38.5.13.6 C(n)V and C(n+1)V register synchronization

The C(n)V and C(n+1)V registers synchronization updates the C(n)V and C(n+1)V registers with their buffer values.

This synchronization is enabled if (FTMEN = 1) and (SYNCEN = 1). The synchronization mechanism is the same as the MOD register synchronization. However, it is expected that the C(n)V and C(n+1)V registers be synchronized only by the enhanced PWM synchronization (SYNCMODE = 1).

38.5.13.7 OUTMASK register synchronization

The OUTMASK register synchronization updates the OUTMASK register with its buffer value.

The OUTMASK register can be updated at each rising edge of system clock (SYNCHOM = 0), by the enhanced PWM synchronization (SYNCHOM = 1 and SYNCMODE = 1) or by the legacy PWM synchronization (SYNCHOM = 1 and SYNCMODE = 0). However, it is expected that the OUTMASK register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the OUTMASK register synchronization depends on SWOM and HWOM bits. See the following flowchart:

For Assessment Purposes Only

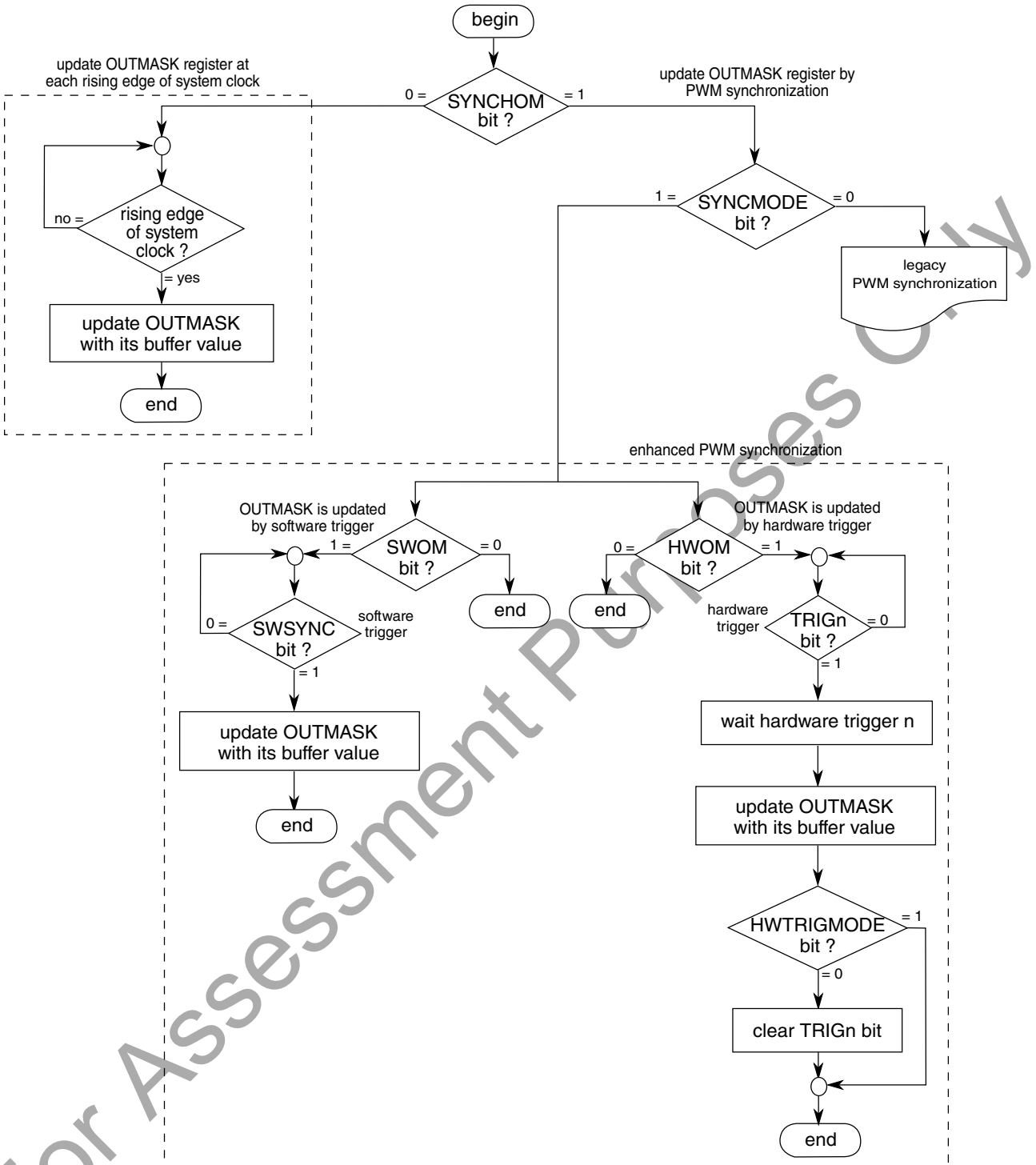


Figure 38-57. OUTMASK register synchronization flowchart

In the case of legacy PWM synchronization, the OUTMASK register synchronization depends on PWMSYNC bit according to the following description.

For Assessment Purposes Only

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 0), then this synchronization is done on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected loading point. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

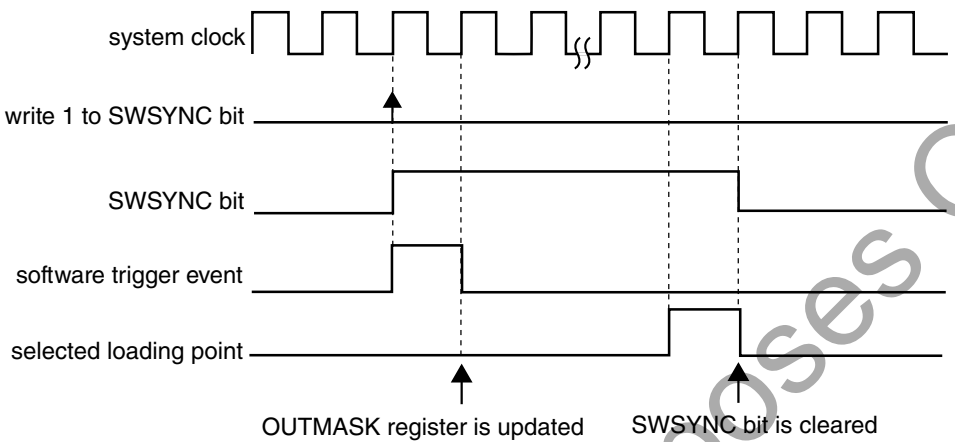


Figure 38-58. OUTMASK synchronization with (SYNCMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0) and software trigger was used

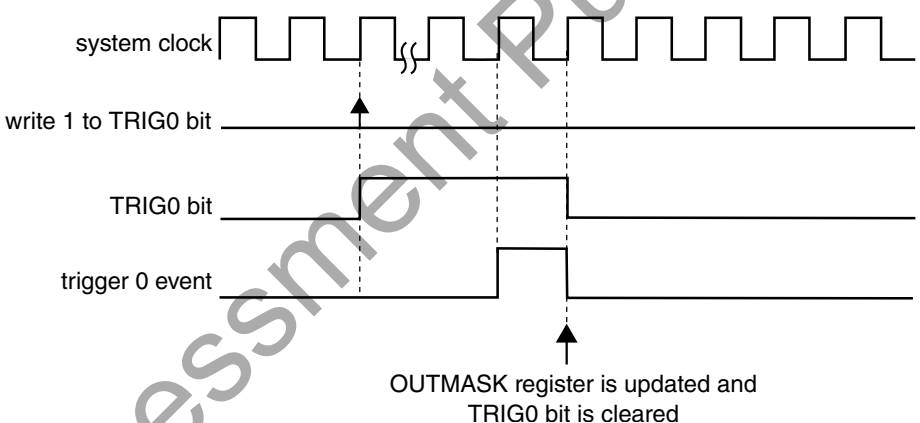


Figure 38-59. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0), and a hardware trigger was used

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 1), then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to Hardware trigger. An example with a hardware trigger follows.

For Assessment Purposes Only

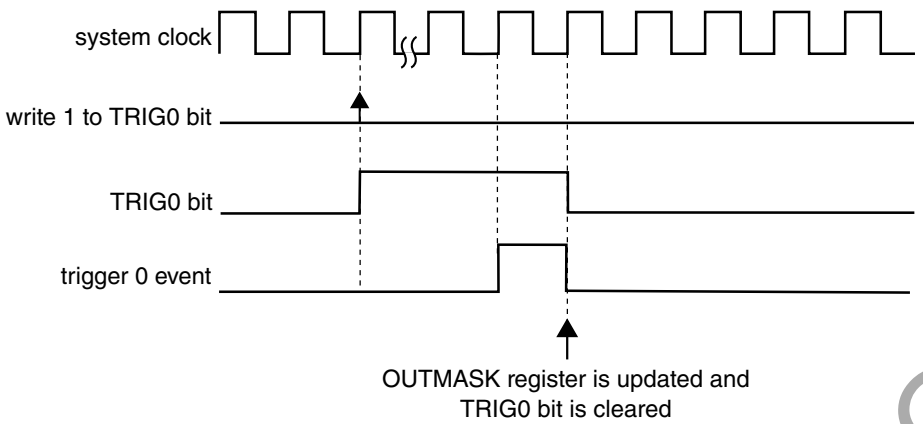


Figure 38-60. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 1), and a hardware trigger was used

38.5.13.8 INVCTRL register synchronization

The INVCTRL register synchronization updates the INVCTRL register with its buffer value.

The INVCTRL register can be updated at each rising edge of system clock (INVC = 0) or by the enhanced PWM synchronization (INVC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the INVCTRL register synchronization depends on SWINVC and HWINVC bits.

For Assessment Purposes Only

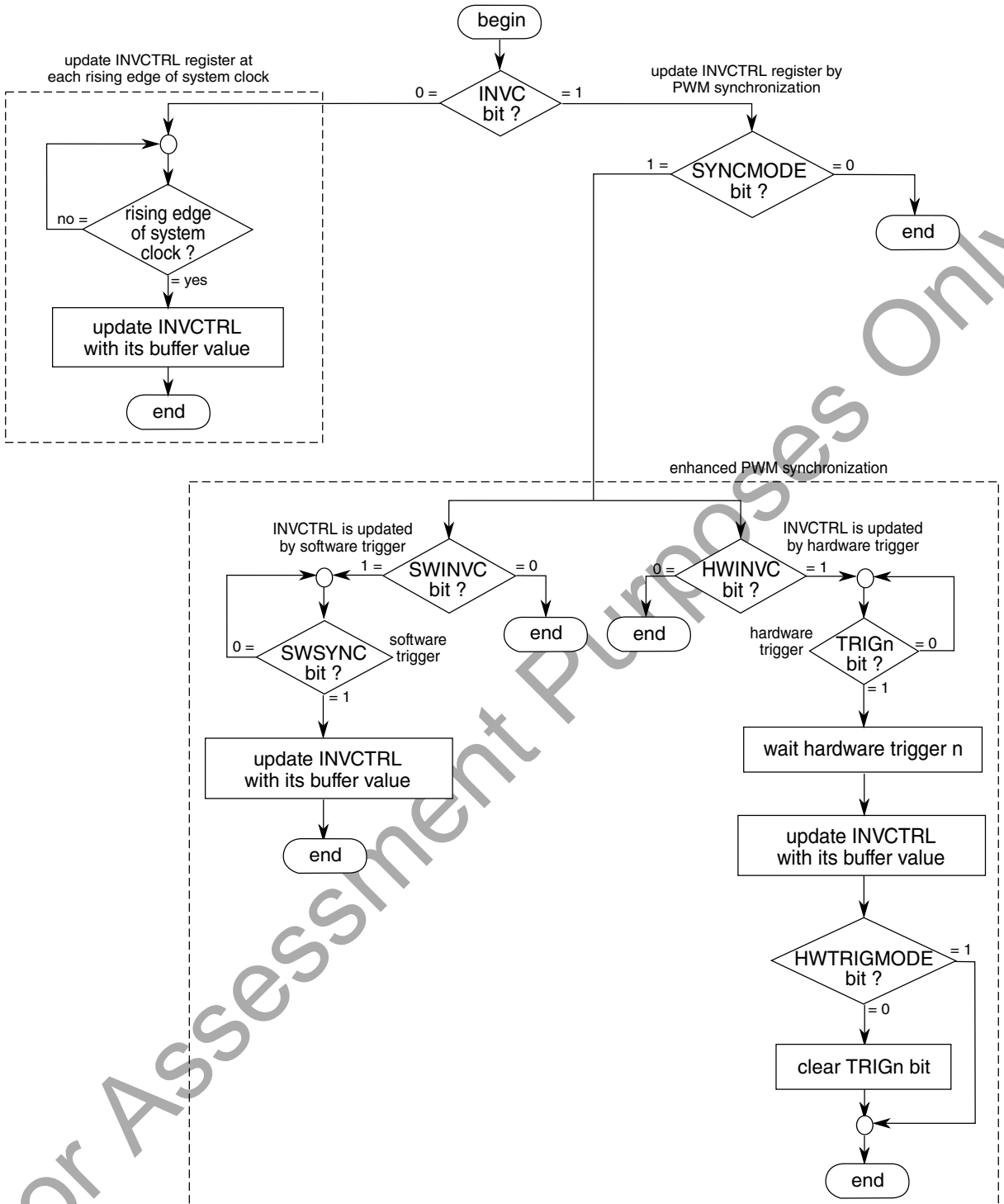


Figure 38-61. INVCTRL register synchronization flowchart

For Assessment Purposes Only

38.5.13.9 SWOCTRL register synchronization

The SWOCTRL register synchronization updates the SWOCTRL register with its buffer value.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The SWOCTRL register can be updated at each rising edge of system clock (SWOC = 0) or by the enhanced PWM synchronization (SWOC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the SWOCTRL register synchronization depends on SWSOC and HWSOC bits.

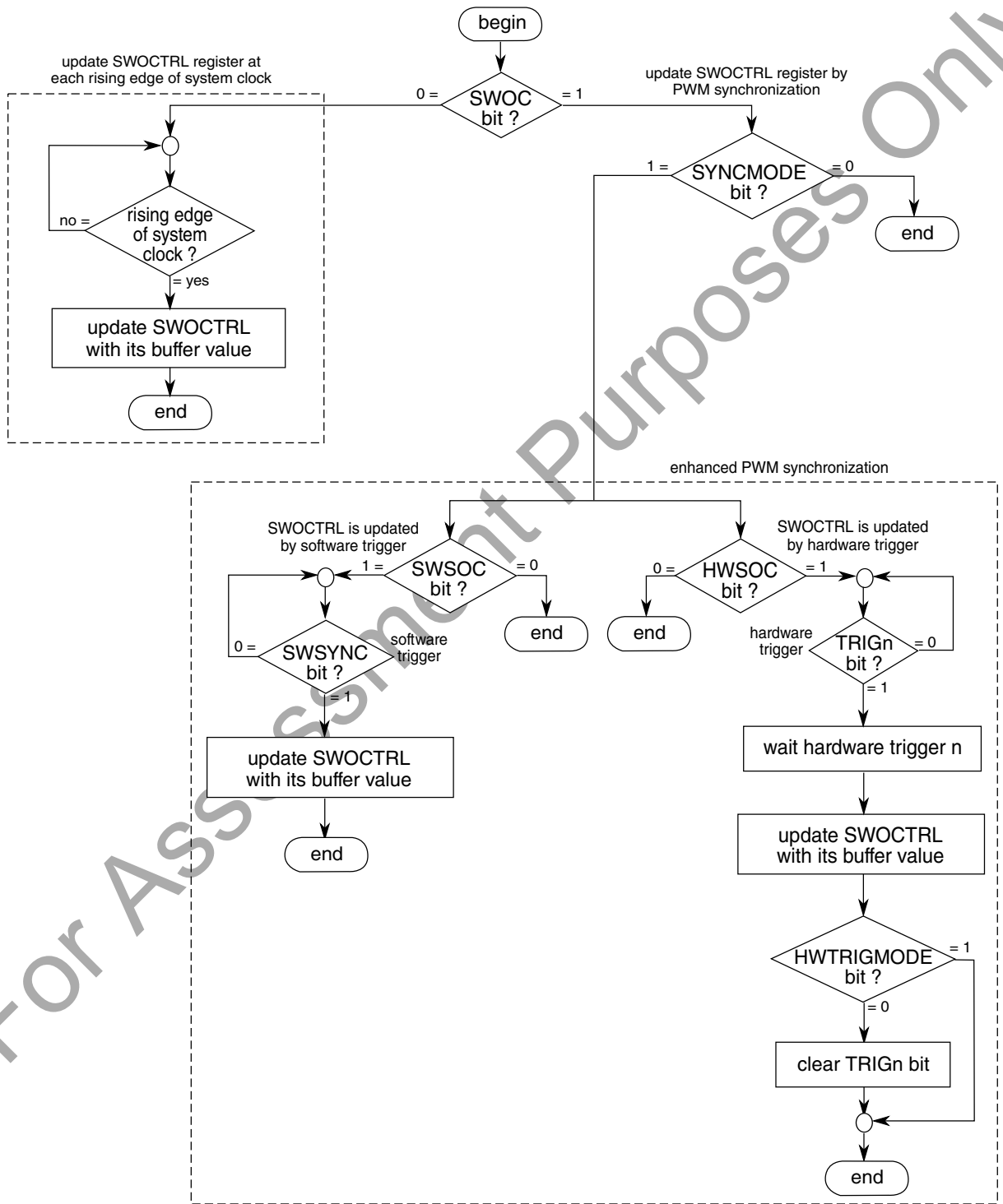


Figure 38-62. SWOCTRL register synchronization flowchart

For Assessment Purposes Only

38.5.13.10 FTM counter synchronization

The FTM counter synchronization is a mechanism that allows the FTM to restart the PWM generation at a certain point in the PWM period. The channels outputs are forced to their initial value, except for channels in Output Compare mode, and the FTM counter is forced to its initial counting value defined by CNTIN register.

The following figure shows the FTM counter synchronization. Note that after the synchronization event occurs, the channel (n) is set to its initial value and the channel (n +1) is not set to its initial value due to a specific timing of this figure in which the deadtime insertion prevents this channel output from transitioning to 1. If no deadtime insertion is selected, then the channel (n+1) transitions to logical value 1 immediately after the synchronization event occurs.

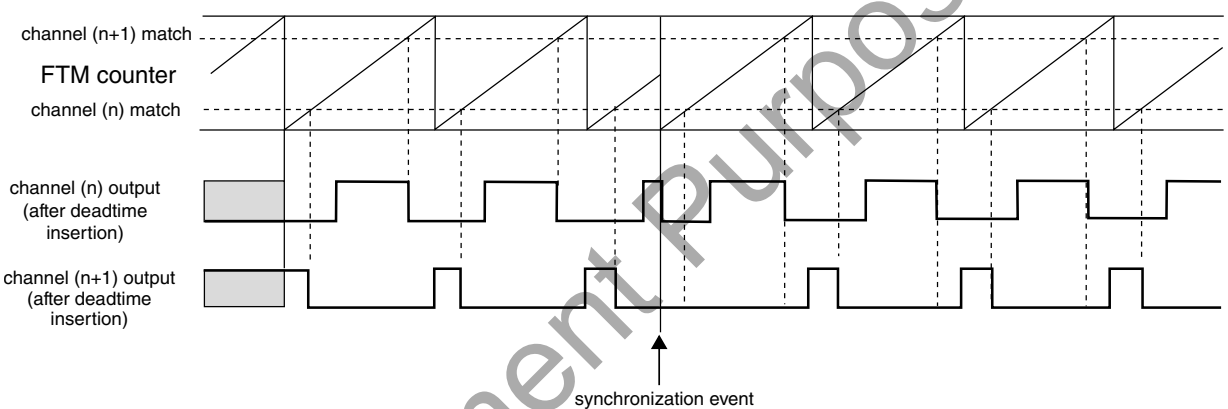


Figure 38-63. FTM counter synchronization

The FTM counter synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, the FTM counter must be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the FTM counter synchronization depends on SWRSTCNT and HWRSTCNT bits according to the following flowchart.

For Assessment Purposes Only

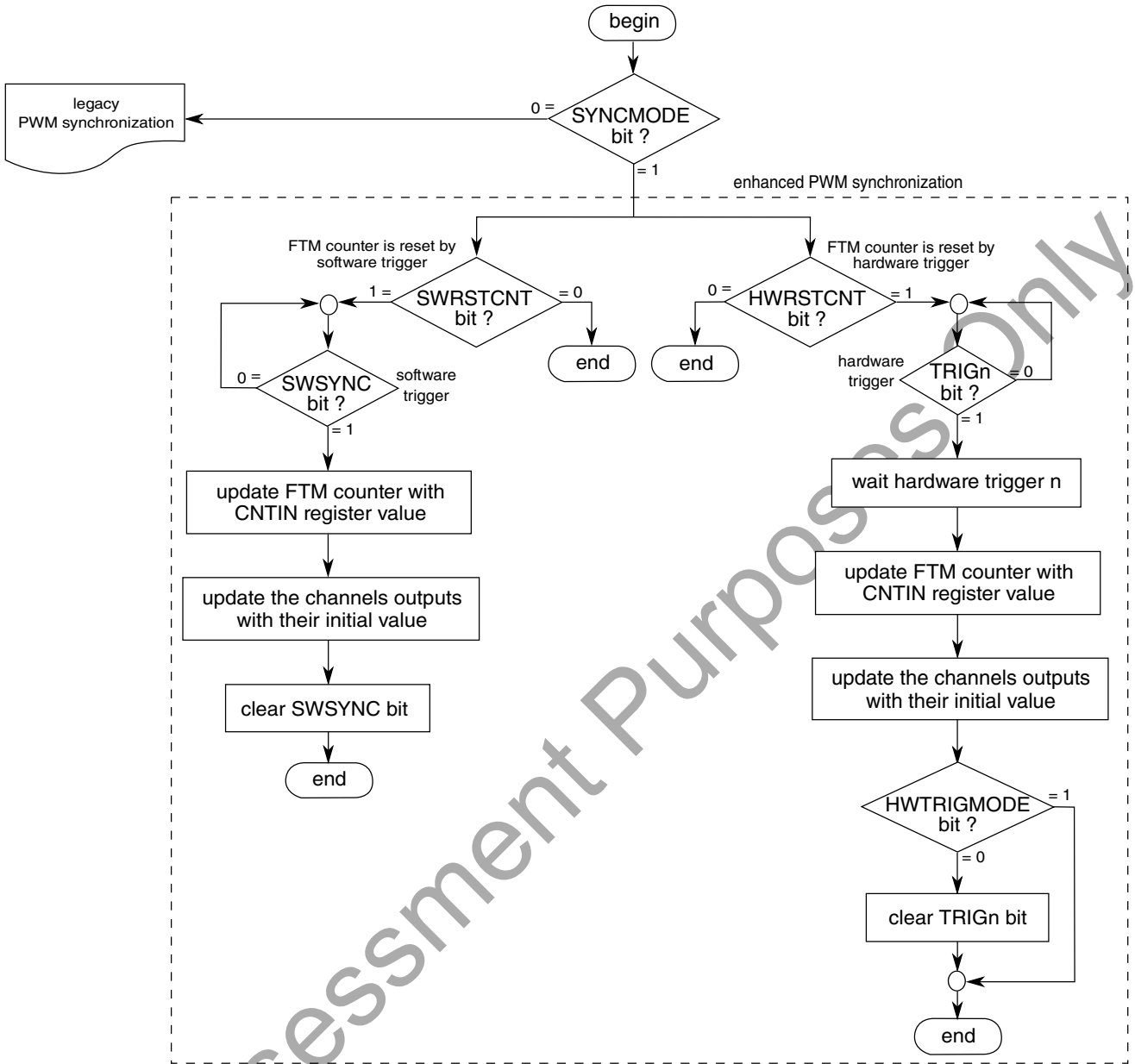


Figure 38-64. FTM counter synchronization flowchart

In the case of legacy PWM synchronization, the FTM counter synchronization depends on REINIT and PWMSYNC bits according to the following description.

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 0) then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

For Assessment Purposes Only

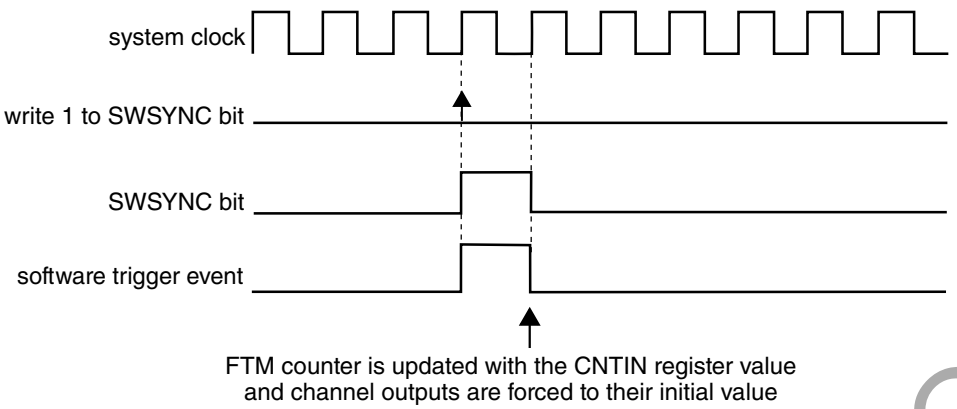


Figure 38-65. FTM counter synchronization with (SYNCMODE = 0), (REINIT = 1), (PWMSYNC = 0), and software trigger was used

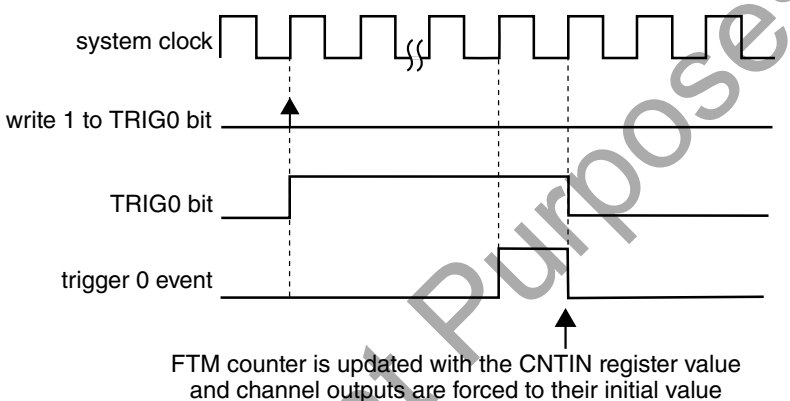


Figure 38-66. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 0), and a hardware trigger was used

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 1) then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to Hardware trigger.

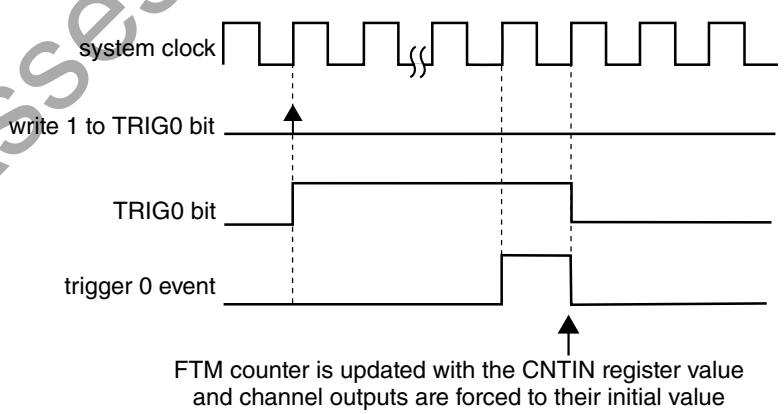


Figure 38-67. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 1), and a hardware trigger was used

For Assessment Purposes Only

38.5.14 Half and full cycle reload

The PWMLOAD register allows to update the MOD, HCR, CNTIN, and C(n)V registers with the content of the register buffer at a defined reload opportunity. In this case, it is not required to use the PWM synchronization.

There are multiple possible reload opportunities for registers reload. Each reload opportunity can turn into a reload point or not. It will depends on load frequency counter. If, for example, the load frequency is chosen to zero, then any reload opportunity is also a reload point. Note that when a reload point is reached, a register reload will only occur if LDOK bit is enabled. The reload flag (RF) and initialization trigger generation are independent of LDOK bit. The table below shows which are the reload opportunities.

Table 38-11. When possible reload opportunities are enabled

Loading point	Enabled
When a counter event happens. See Counter events.	Always
At the channel (j) match (FTM counter = C(j)V)	When CHjSEL = 1
At the Half cycle event match (FTM counter = HCR)	When HCSEL = 1

The figure below shows a simplified representation of the reload logic. The Reload Flag (RF) can be used to generate an external interrupt when a load point is reached. It is also possible to generate an initialization trigger and a register reload when a load point is reached. Note that Load Frequency configuration can modify the RF generation.

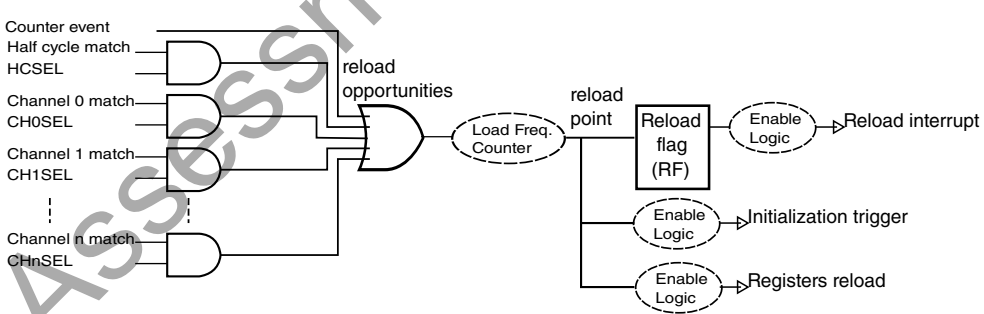


Figure 38-68. Registers reload logic

The following figure shows some examples of enabled reload opportunities when counter is in up counting mode. Note that the example below also uses a channel match as reload opportunity, but generally applications uses only the half cycle match if a non full cycle reload is needed.

For Assessment Purposes Only

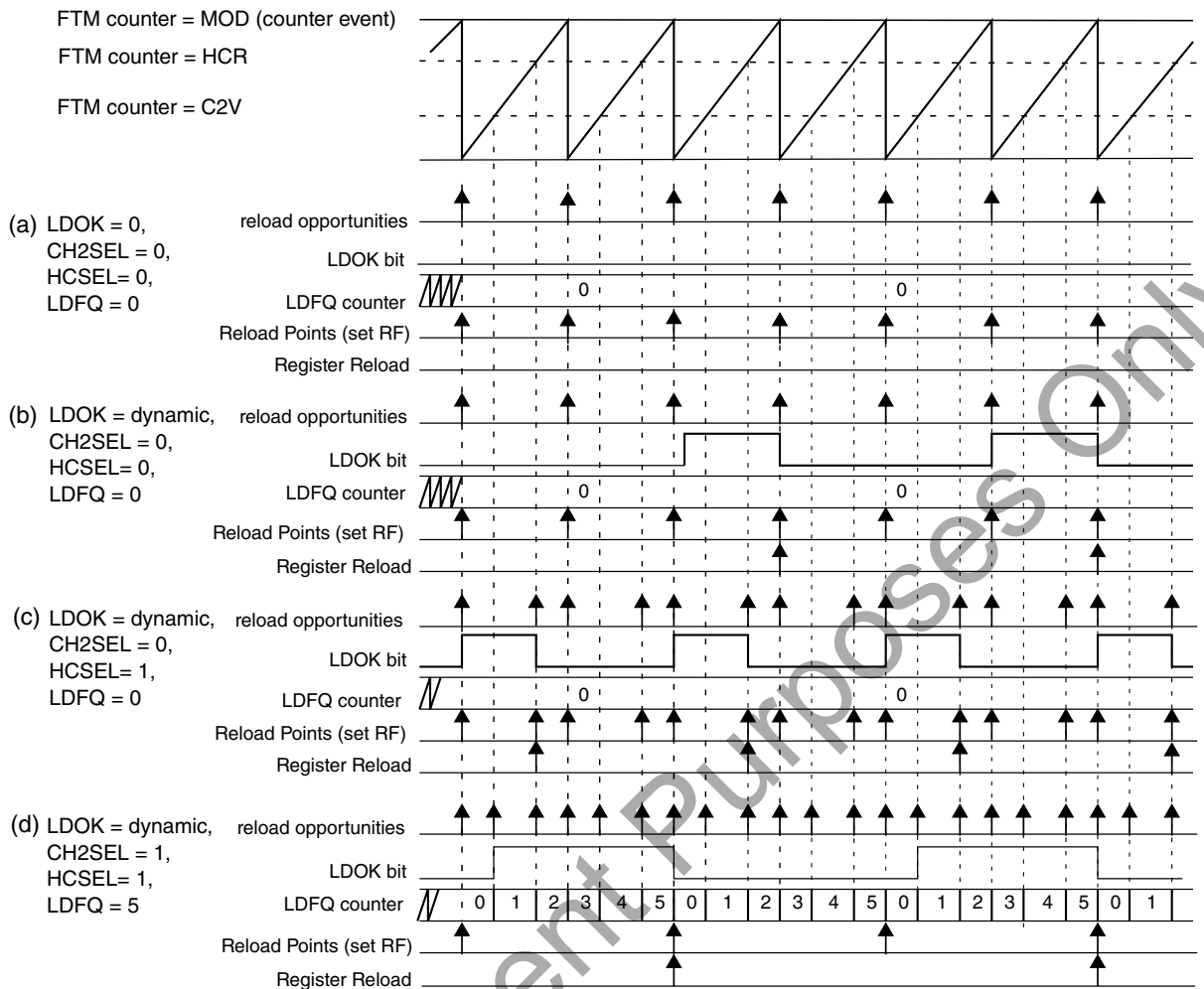


Figure 38-69. Reload opportunities to half and full cycle reload when up counting

The table below shows the possible counter events selection (reload opportunities) to up-down counting mode:

Table 38-12. Reload opportunities to up-down counting mode

FTM_SYNC bits	Reload opportunities selected
CNTMIN = 0 and CNTMAX = 0	When the counter turns from up to down (compatibility mode).
CNTMIN = 1 and CNTMAX = 0	When the counter turns from down to up.
CNTMIN = 0 and CNTMAX = 1	When the counter turns from up to down.
CNTMIN = 1 and CNTMAX = 1	When the counter turns from down to up and When the counter turns from up to down.

After enabling the reload opportunities, the LDOK bit must be set for the reload to occur. In this case, the reload occurs at the next enabled reload point considering the Load Frequency according to the following conditions:

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 38-13. Conditions for reload occurring at the next enabled reload point

When a new value was written	Then
To the MOD register	The MOD register is updated with its write buffer value.
To the HCR register	The HCR register is updated with its write buffer value.
To the CNTIN register and CNTINC = 1	The CNTIN register is updated with its write buffer value.
To the C(n)V register and SYNCENm = 1 – where m indicates the pair channels (n) and (n+1)	The C(n)V register is updated with its write buffer value.
To the C(n+1)V register and SYNCENm = 1 – where m indicates the pair channels (n) and (n+1)	The C(n+1)V register is updated with its write buffer value.

NOTE

- If ELSjB and ELSjA bits are different from zero, then the channel (j) output signal is generated according to the configured output mode. If ELSjB and ELSjA bits are zero, then the generated signal is not available on channel (j) output.
- If CHjIE = 1, then the channel (j) interrupt is generated when the channel (j) match occurs.
- At these reload points neither the channels outputs nor the FTM counter are changed. Software must select these reload points at the safe points in time.

38.5.15 Inverting

The invert functionality swaps the signals between channel (n) and channel (n+1) outputs. The inverting operation is selected when:

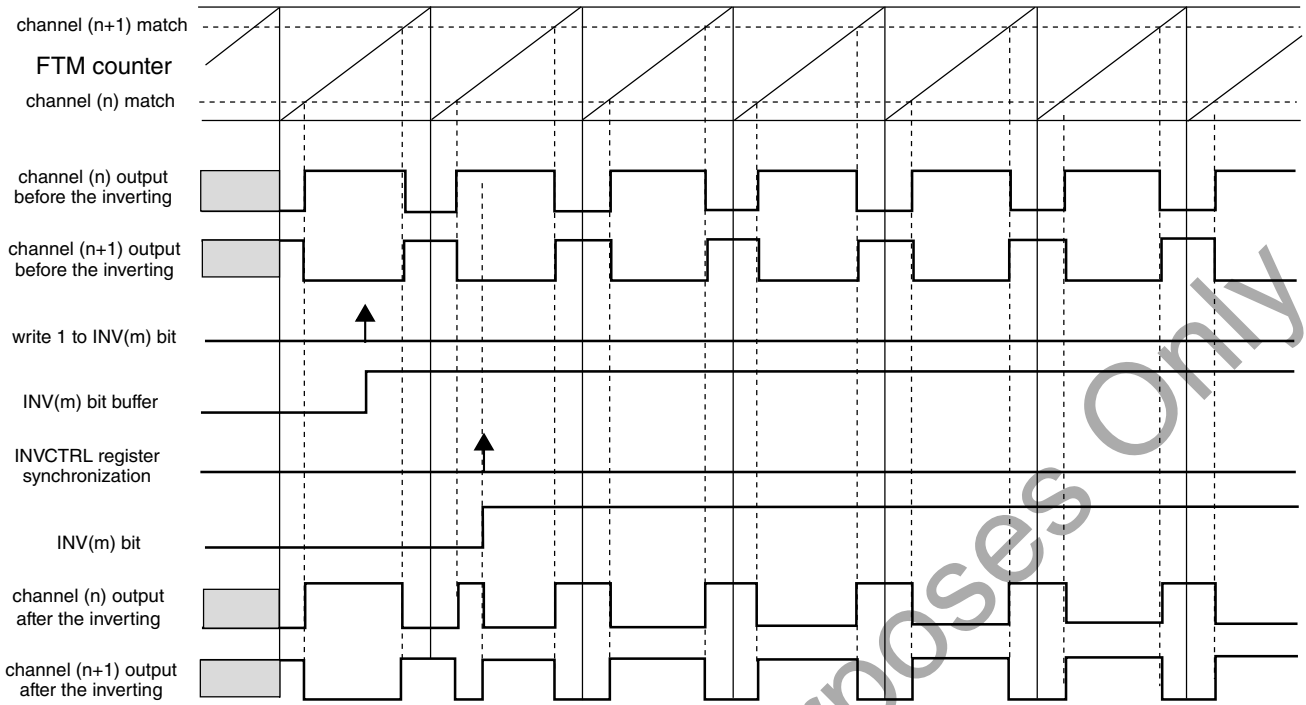
- QUADEN = 0
- DECAPEN = 0
- COMP = 1, and
- INVm = 1 (where m represents a channel pair)

The INVm bit in INVCTRL register is updated with its buffer value according to INVCTRL register synchronization

In High-True (ELSnB:ELSnA = 1:0) Combine mode, the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN), forced high at the channel (n) match and forced low at the channel (n+1) match. If the inverting is selected, the channel (n) output behavior is changed to force high at the beginning of the PWM period, force low at the channel (n) match and force high at the channel (n+1) match. See the following figure.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

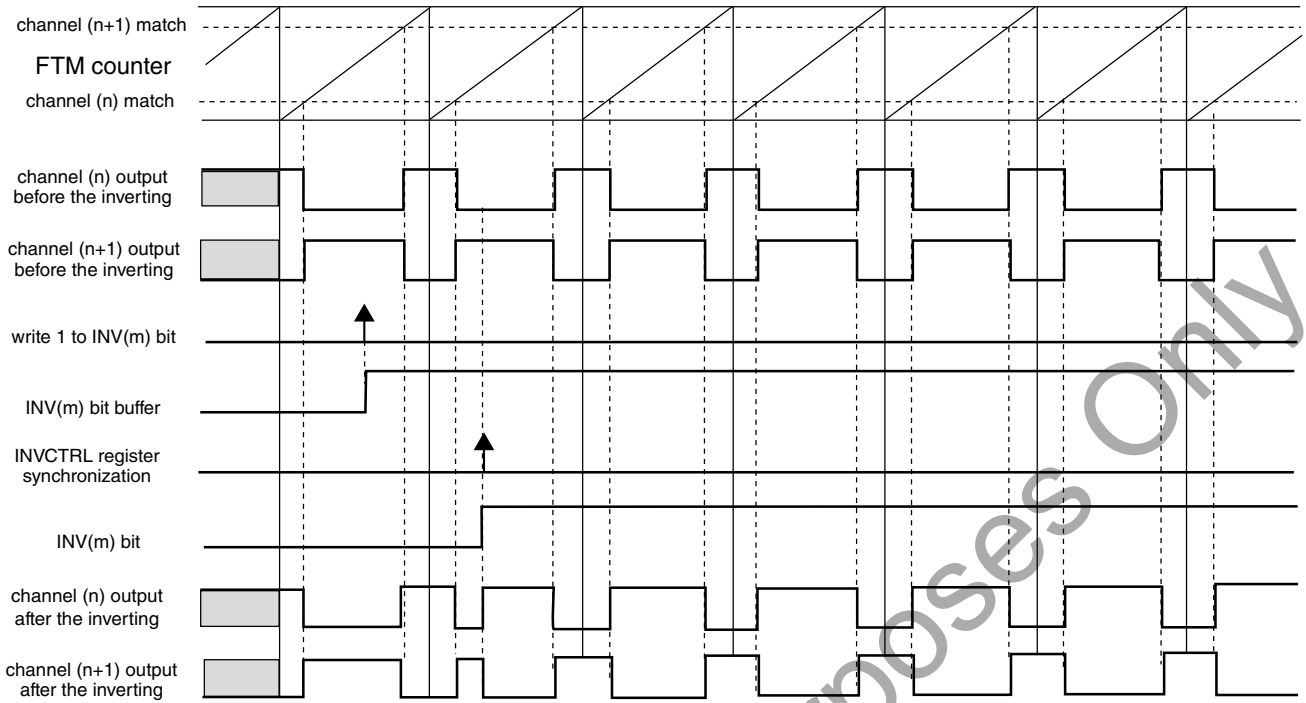


NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Figure 38-70. Channels (n) and (n+1) outputs after the inverting in High-True (ELSnB:ELSnA = 1:0) Combine mode

Note that the ELSnB:ELSnA bits value should be considered because they define the active state of the channels outputs. In Low-True (ELSnB:ELSnA = X:1) Combine mode, the channel (n) output is forced high at the beginning of the period, forced low at the channel (n) match and forced high at the channel (n+1) match. When inverting is selected, the channels (n) and (n+1) present waveforms as shown in the following figure.

For Assessment Purposes Only



NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Figure 38-71. Channels (n) and (n+1) outputs after the inverting in Low-True (ELSnB:ELSnA = X:1) Combine mode

NOTE

The Inverting is not available in Output Compare mode and Modified Combine PWM Mode.

38.5.16 Software Output Control Mode

The software output control forces the channel output according to software defined values at a specific time in the PWM generation.

The software output control is selected when:

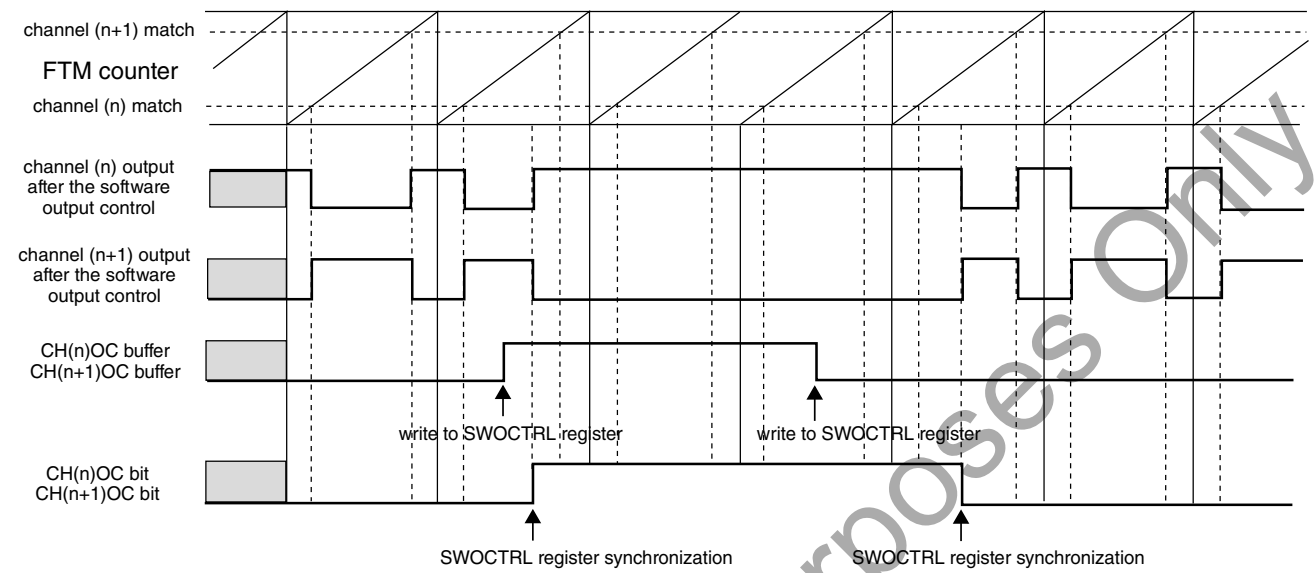
- QUADEN = 0
- DECAPEN = 0, and
- CHnOC = 1

The CHnOC bit enables the software output control for a specific channel output and the CHnOCV selects the value that is forced to this channel output.

Both CHnOC and CHnOCV bits in SWOCTRL register are buffered and updated with their buffer value according to SWOCTRL register synchronization.

For Assessment Purposes Only

The following figure shows the channels (n) and (n+1) outputs signals when the software output control is used. In this case the channels (n) and (n+1) are set to Combine and Complementary mode.



NOTE
CH(n)OCV = 1 and CH(n+1)OCV = 0.

Figure 38-72. Example of software output control in Combine and Complementary mode

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is zero.

Table 38-14. Software ouput control behavior when (COMP = 0)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to one

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is one.

Table 38-15. Software ouput control behavior when (COMP = 1)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to zero

For Assessment Purposes Only

Note

- The CH(n)OC and CH(n+1)OC bits should be equal.
- The COMP bit must not be modified when software output control is enabled, that is, CH(n)OC = 1 and/or CH(n +1)OC = 1.
- Software output control has the same behavior with disabled or enabled FTM counter (see the CLKS field description in the Status and Control register).

38.5.17 Deadtime insertion

The deadtime insertion is enabled when DTEN is set and the concatenation {DTVALEX[3:0],DTVAL[5:0]} is non- zero.

DEADTIME register defines the deadtime delay that can be used for all FTM channels. The clock for the DEADTIME delay is the system clock divided by DTPS bits and the {DTVALEX[3:0],DTVAL[5:0]} bits define the deadtime modulo, that is, the number of the deadtime prescaler clocks.

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) = 0, and the deadtime is enabled, then when the channel (n) match (FTM 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 (FTM 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) = 1, and the deadtime is enabled, then when the channel (n) match (FTM 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 (FTM 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

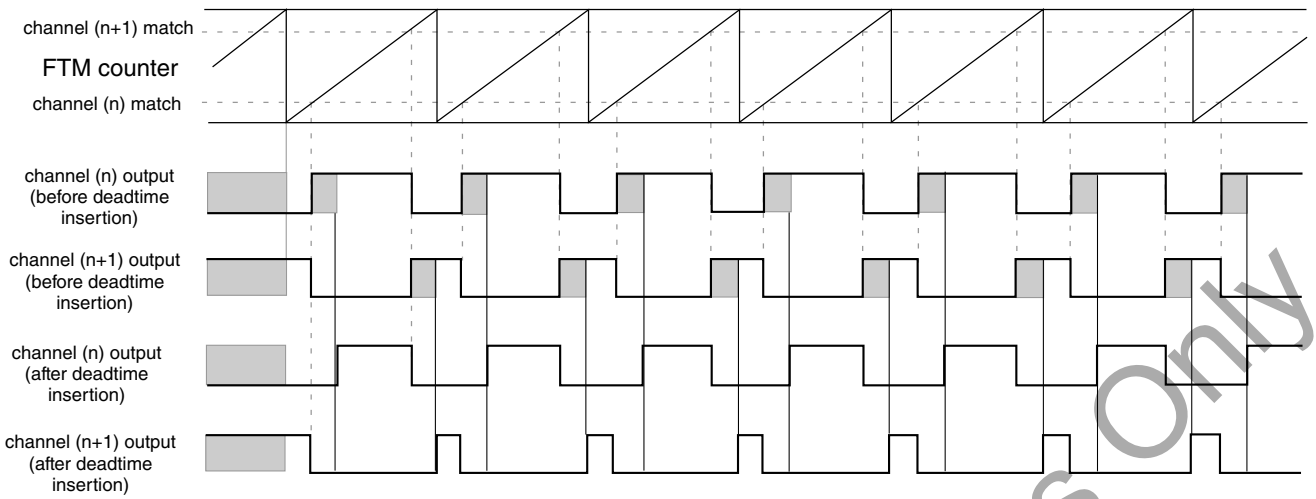


Figure 38-73. Deadtime insertion with ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0

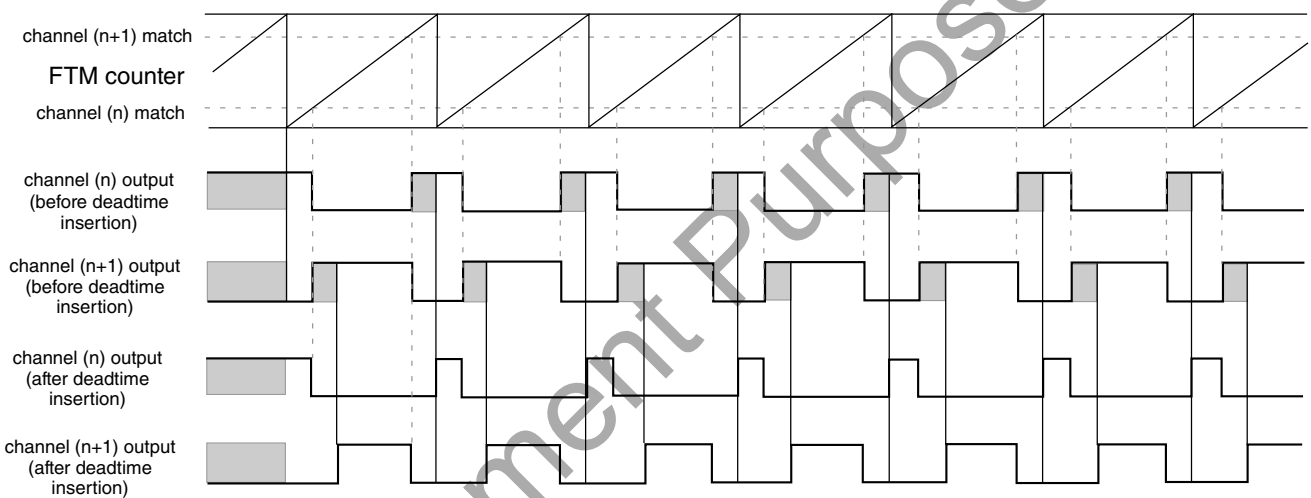


Figure 38-74. Deadtime insertion with ELSnB:ELSnA = X:1, POL(n) = 0, and POL(n+1) = 0

NOTE

- The deadtime feature must be used only in Complementary mode.
- The deadtime feature is not available in Output Compare mode.

38.5.17.1 Deadtime insertion corner cases

If (PS[2:0] is cleared), (DTPS[1:0] = 0:0 or DTPS[1:0] = 0:1):

For Assessment Purposes Only

- and the deadtime delay is greater than or equal to the channel (n) duty cycle ((C(n +1)V – C(n)V) × system clock), then the channel (n) output is always the inactive value (POL(n) bit value).
- and the deadtime delay is greater than or equal to the channel (n+1) duty cycle ((MOD – CNTIN + 1 – (C(n+1)V – C(n)V)) × system clock), then the channel (n+1) output is always the inactive value (POL(n+1) bit value).

Although, in most cases the deadtime delay is not comparable to channels (n) and (n+1) duty cycle, the following figures show examples where the deadtime delay is comparable to the duty cycle.

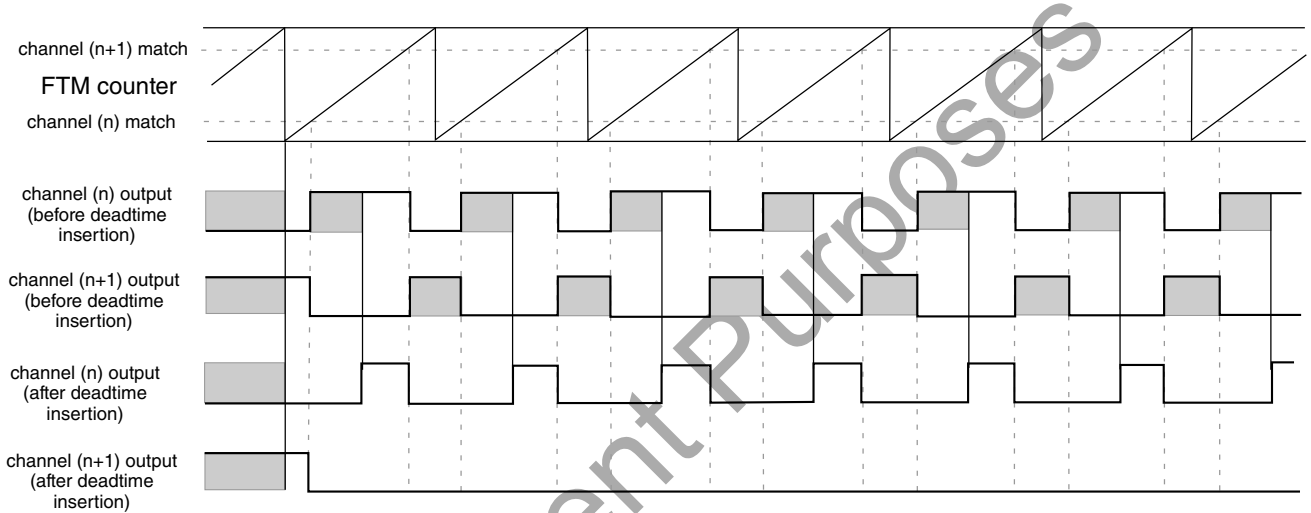


Figure 38-75. Example of the deadtime insertion (ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0) when the deadtime delay is comparable to channel (n+1) duty cycle

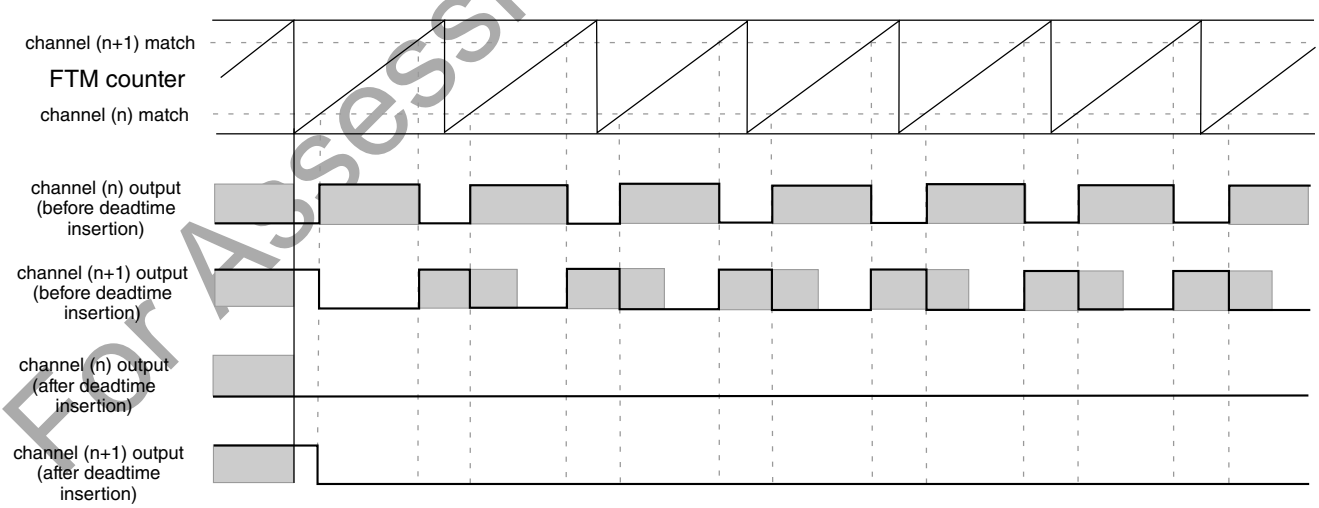


Figure 38-76. Example of the deadtime insertion (ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0) when the deadtime delay is comparable to channels (n) and (n+1) duty cycle

For Assessment Purposes Only

38.5.18 Output mask

The output mask can be used to force channels output to their inactive state through software. For example: to control a BLDC motor.

Any write to the OUTMASK register updates its write buffer. The OUTMASK register is updated with its buffer value by PWM synchronization; see OUTMASK register synchronization.

If CHnOM = 1, then the channel (n) output is forced to its inactive state (POLn bit value). If CHnOM = 0, then the channel (n) output is unaffected by the output mask. See the following figure.

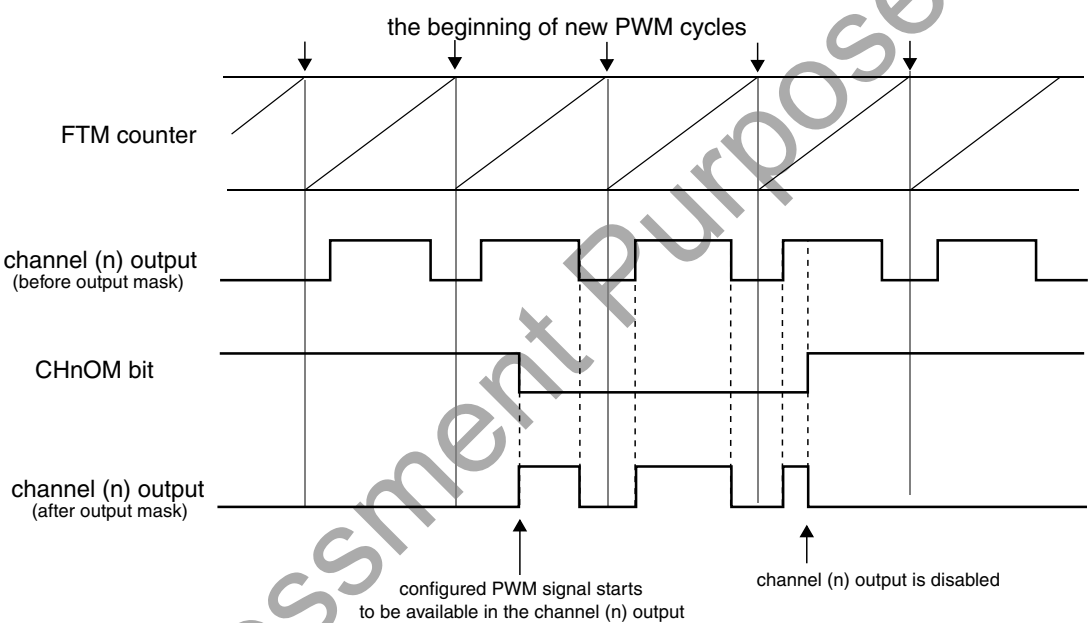


Figure 38-77. Output mask with POLn = 0

The following table shows the output mask result before the polarity control.

Table 38-16. Output mask result for channel (n) before the polarity control

CHnOM	Output Mask Input	Output Mask Result
0	inactive state	inactive state
	active state	active state
1	inactive state	inactive state
	active state	

Assessment Purposes Only

38.5.19 Fault control

The fault control is enabled if (FAULTM[1:0] ≠ 0:0).

FTM can have up to four fault inputs. FAULTnEN bit (where n = 0, 1, 2, 3) enables the fault input n and FFLTRnEN bit enables the fault input n filter. FFVAL[3:0] bits select the value of the enabled filter in each enabled fault input.

First, each fault input signal is synchronized by the system clock; see the synchronizer block in the following figure. Following synchronization, the fault input n signal enters the filter block. The filter clock is prescaled and controlled by FLTPS[3:0] bits. When there is a state change in the fault input n signal, the 5-bit counter is reset and starts counting up. As long as the new state is stable on the fault input n, the counter continues to increment. If the 5-bit counter overflows, that is, the counter exceeds the value of the FFVAL[3:0] bits, the new fault input n value is validated. It is then transmitted as a pulse edge to the edge detector.

If the opposite edge appears on the fault input n signal before validation (counter overflow), 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 FFVAL[3:0] bits (× filter clock) is regarded as a glitch and is not passed on to the edge detector.

The fault input n filter is disabled when the FFVAL[3:0] bits are zero or when FAULTnEN = 0. In this case, the fault input n signal is delayed 2 rising edges of the system clock and the FAULTFn bit is set on 3th rising edge of the system clock after a rising edge occurs on the fault input n.

If FFVAL[3:0] ≠ 0000 and FAULTnEN = 1, then the delay is (2 system clocks + (2 + FFVAL) filter clk), that is, the FAULTFn bit is set (2 system clocks + 2 filter clocks + FFVAL[3:0]) clock edges after a rising edge occurs on the fault input n.

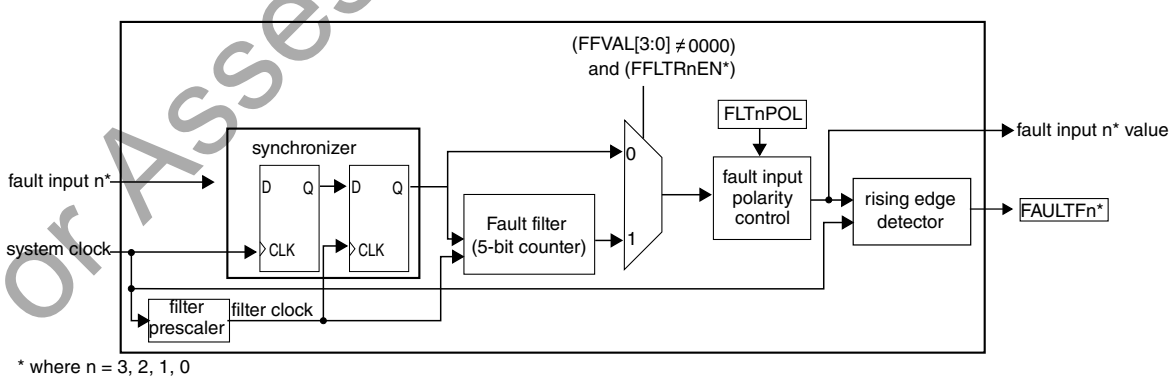


Figure 38-78. Fault input n control block diagram

If the fault control and fault input n are enabled and a rising edge at the fault input n signal is detected, a fault condition has occurred and the FAULTFn bit is set. The FAULTF bit is the logic OR of FAULTFn[3:0] bits. See the following figure.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

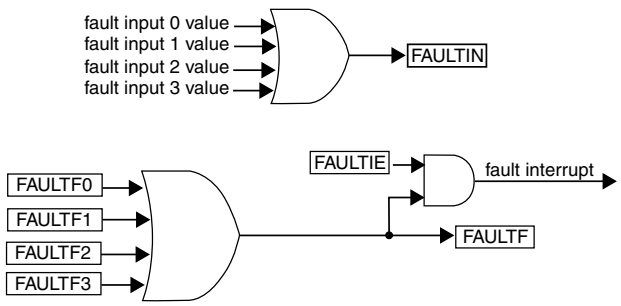


Figure 38-79. FAULTF and FAULTIN bits and fault interrupt

If the fault control is enabled (FAULTM[1:0] ≠ 0:0), a fault condition has occurred and (FAULTEN = 1), then outputs are forced to their safe values:

- Channel (n) output takes the value of POL(n)
- Channel (n+1) takes the value of POL(n+1)

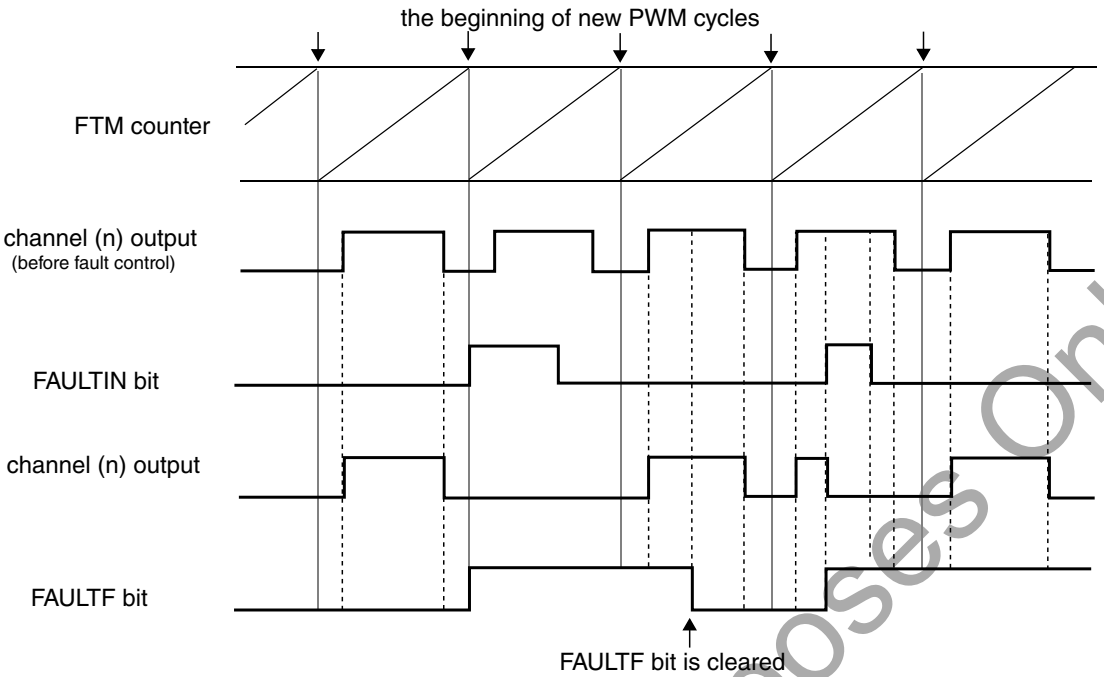
The fault interrupt is generated when (FAULTF = 1) and (FAULTIE = 1). This interrupt request remains set until:

- Software clears the FAULTF bit by reading FAULTF bit as 1 and writing 0 to it
- Software clears the FAULTIE bit
- A reset occurs

38.5.19.1 Automatic fault clearing

If the automatic fault clearing is selected (FAULTM[1:0] = 1:1), then the channels output disabled by fault control is again enabled when the fault input signal (FAULTIN) returns to zero and a new PWM cycle begins. See the following figure.

For Assessment Purposes Only



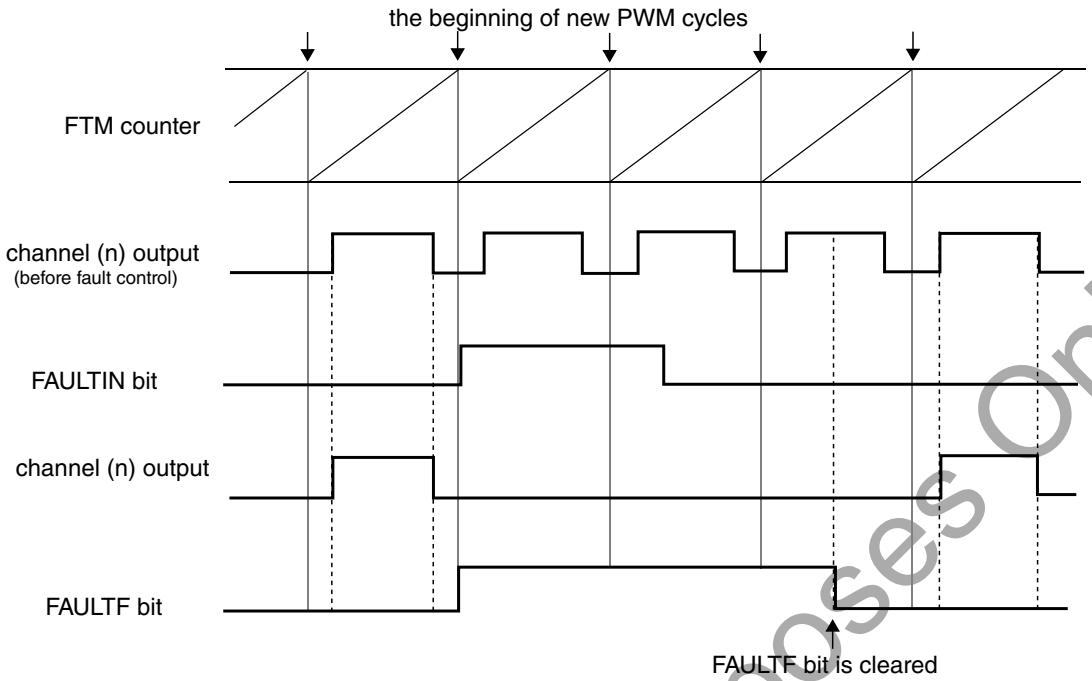
NOTE
The channel (n) output is after the fault control with automatic fault clearing and POLn = 0.

Figure 38-80. Fault control with automatic fault clearing

38.5.19.2 Manual fault clearing

If the manual fault clearing is selected (FAULTM[1:0] = 0:1 or 1:0), then the channels output disabled by fault control is again enabled when the FAULTF bit is cleared and a new PWM cycle begins. See the following figure.

For Assessment Purposes Only



NOTE
The channel (n) output is after the fault control with manual fault clearing and POLn = 0.

Figure 38-81. Fault control with manual fault clearing

38.5.19.3 Fault inputs polarity control

The FLTjPOL bit selects the fault input j polarity, where j = 0, 1, 2, 3:

- If FLTjPOL = 0, the fault j input polarity is high, so the logical one at the fault input j indicates a fault.
- If FLTjPOL = 1, the fault j input polarity is low, so the logical zero at the fault input j indicates a fault.

38.5.20 Polarity Control

The POLn bit selects the channel (n) output polarity:

- If POLn = 0, the channel (n) output polarity is high, so the logical one is the active state and the logical zero is the inactive state.
- If POLn = 1, the channel (n) output polarity is low, so the logical zero is the active state and the logical one is the inactive state.

For Assessment Purposes Only

38.5.21 Initialization

The initialization forces the CHnOI bit value to the channel (n) output when a one is written to the INIT bit.

The initialization depends on COMP and DTEN bits. The following table shows the values that channels (n) and (n+1) are forced by initialization when the COMP and DTEN bits are zero.

Table 38-17. Initialization behavior when (COMP = 0 and DTEN = 0)

CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	0	is forced to zero	is forced to zero
0	1	is forced to zero	is forced to one
1	0	is forced to one	is forced to zero
1	1	is forced to one	is forced to one

The following table shows the values that channels (n) and (n+1) are forced by initialization when (COMP = 1) or (DTEN = 1).

Table 38-18. Initialization behavior when (COMP = 1 or DTEN = 1)

CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	X	is forced to zero	is forced to one
1	X	is forced to one	is forced to zero

Note

The initialization feature must be used only with disabled FTM counter. See the description of the CLKS field in the Status and Control register.

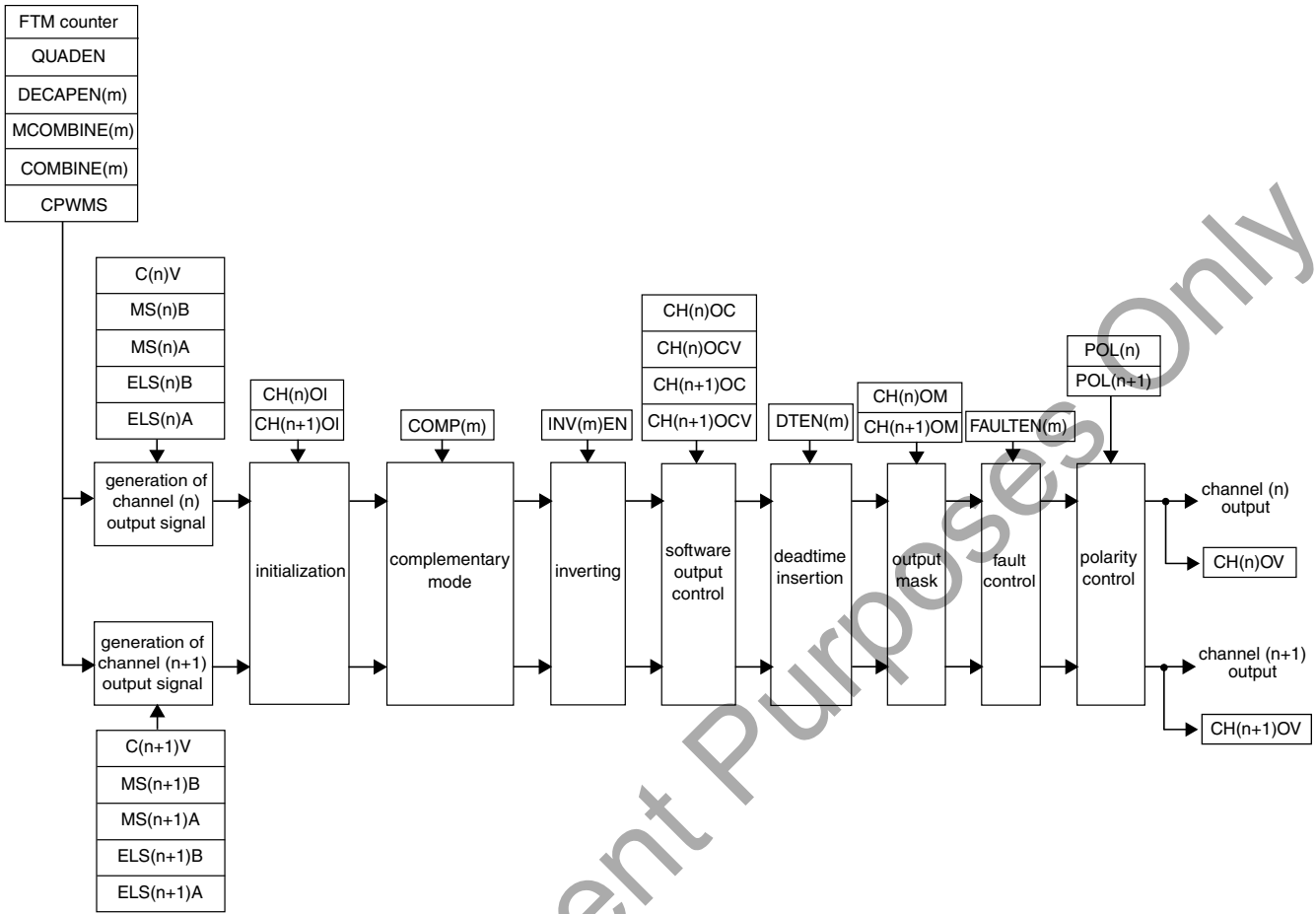
38.5.22 Features priority

The following figure shows the priority of the features used at the generation of channels (n) and (n+1) outputs signals.

For Assessment Purpose Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

pair channels (m) - channels (n) and (n+1)



NOTE
The channels (n) and (n+1) are in Output Compare, EPWM, CPWM, Combine or Modified Combine PWM modes.

Figure 38-82. Priority of the features used at the generation of channels (n) and (n+1) output

NOTE

The Initialization must not be used with Inverting and Software Output Control Mode.

38.5.23 External Trigger

If the CH(j)TRIG bit of the External Trigger (FTM_EXTTRIG) register is set, where j = 0, 1, 2, 3, 4, 5, 6 or 7, then the FTM generates a trigger when the channel (j) match occurs (FTM counter = C(j)V).

The external trigger feature provides a trigger signal which has one FTM clock period width and is used for on-chip modules.

For Assessment Purposes Only

The FTM is able to generate multiple triggers in one PWM period. Because each trigger is generated for a specific channel, several channels are required to implement this functionality. This behavior is described in Figure 38-83.

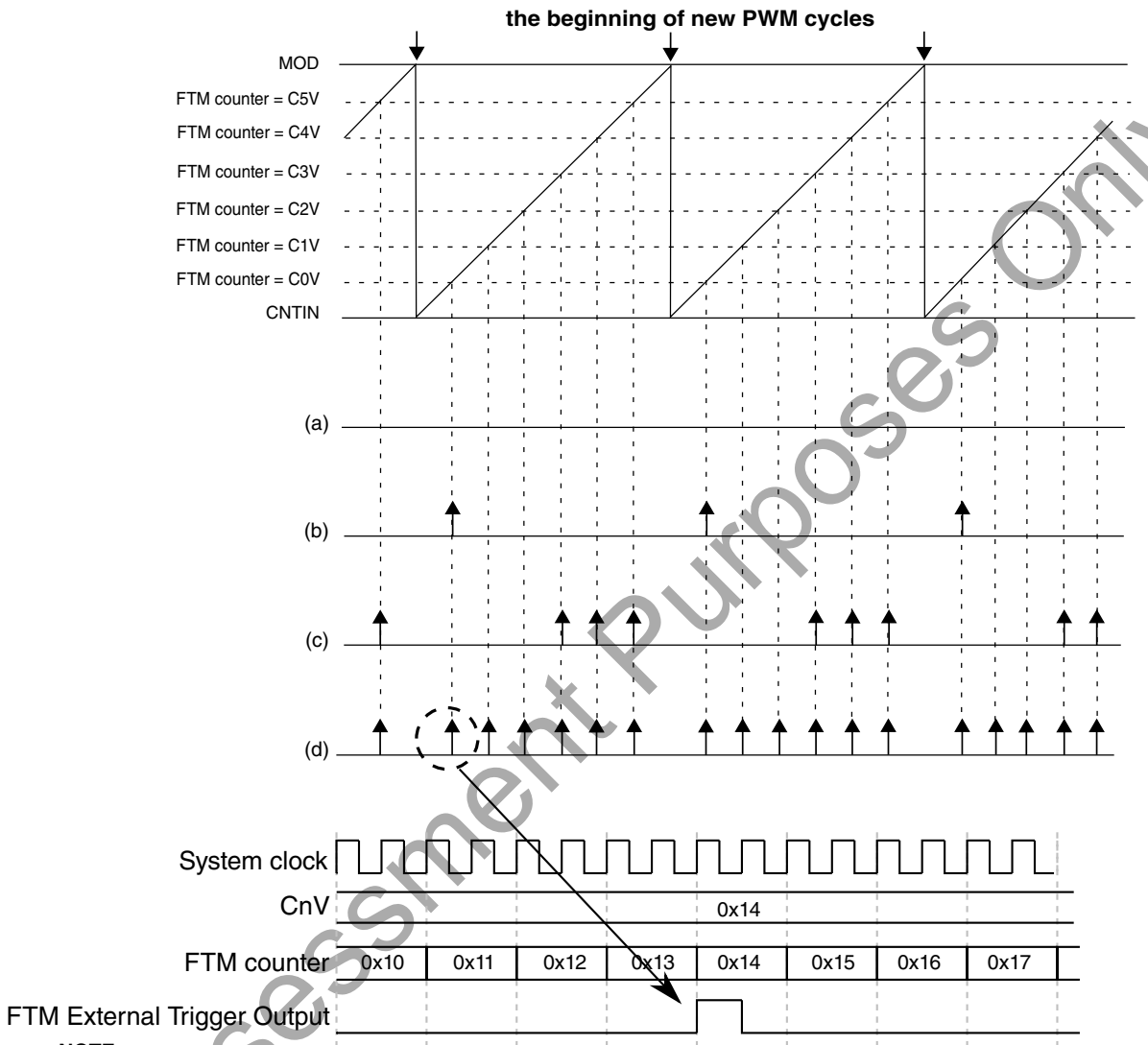


Figure 38-83. External Trigger

For Assessment Purposes Only

38.5.24 Channel trigger output

The channel trigger output provides a trigger signal which has one FTM clock period width in the channel output signal.

If the TRIGMODE bit of the CnSC register is set (TRIGMODE=1), a trigger pulse with one FTM clock cycle width is generated in the channel output when a match occurs. It is only allowed to use trigger mode when channel is in EPWM (up counting) or CPWM (up-down counting).

The figures below show some cases of trigger generation in a channel output.

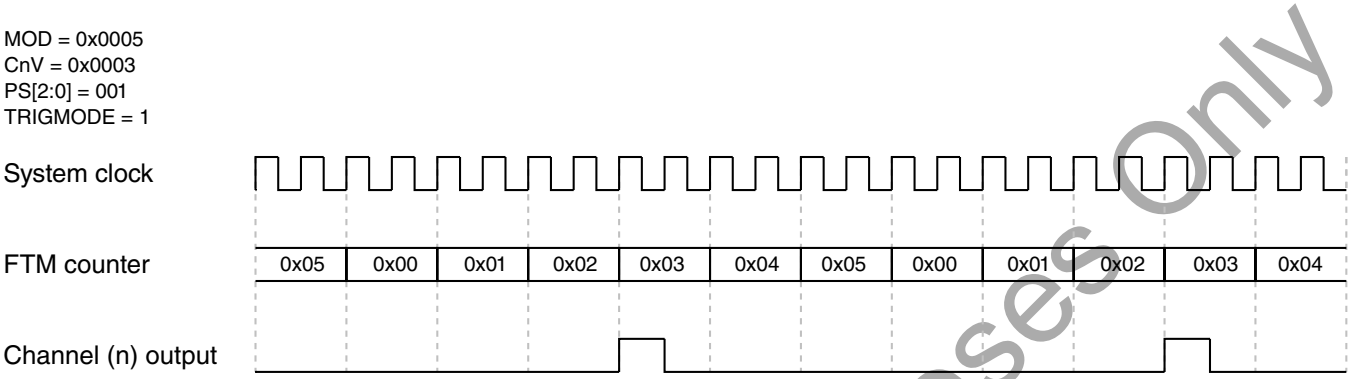


Figure 38-84. Example of trigger generation in the output channel for up counting mode

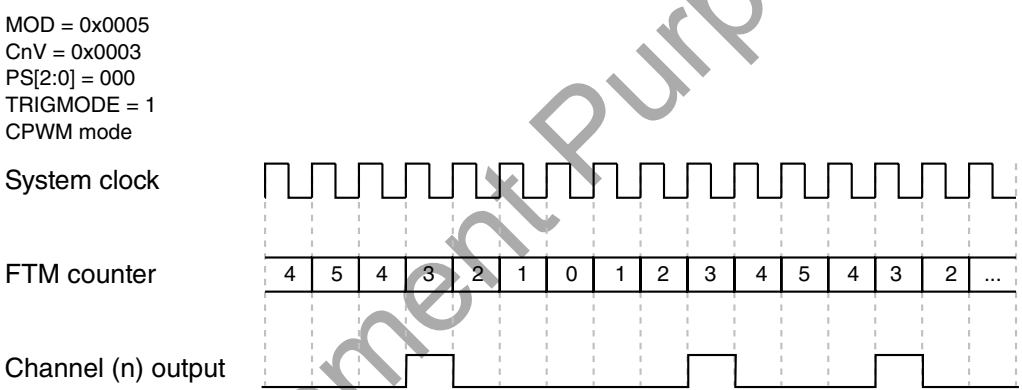


Figure 38-85. Example of trigger generation in the output channel for up-down counting mode

38.5.25 Initialization trigger

Initialization trigger allows FTM to generate an external trigger in some specific points of FTM counter cycle. This feature is controlled by two bits. INITTRIGEN enables the trigger generation and the ITRIGR selects in which events the initialization trigger should be generated. If INITTRIGEN = 1 and ITRIGR = 1, then the initialization trigger is generated when FTM counter reaches a reload point considering the Load Frequency configuration. See the Half and full cycle reload for more details about reload points. If INITTRIGEN = 1 and ITRIGR = 0, then FTM generates a trigger when the FTM counter is updated with the CNTIN register value in the following cases:

- In all cycles that FTM counter is automatically updated with CNTIN register value.

For Assessment Purposes Only

- When there is a write to CNT register.
- When there is the FTM counter synchronization.
- If (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits.
- If the channel (n) is in Input Capture mode, (ICRST = 1) and the selected input capture event occurs in the channel (n) input.

The following figures show these cases.

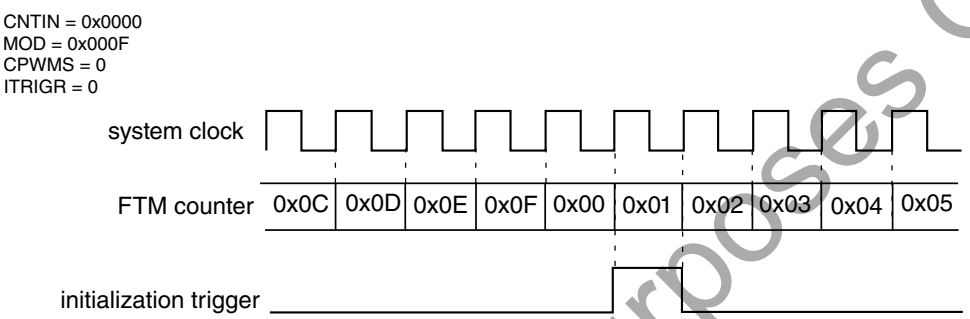


Figure 38-86. Initialization trigger is generated when the FTM counting achieves the CNTIN register value and ITRIGR = 0

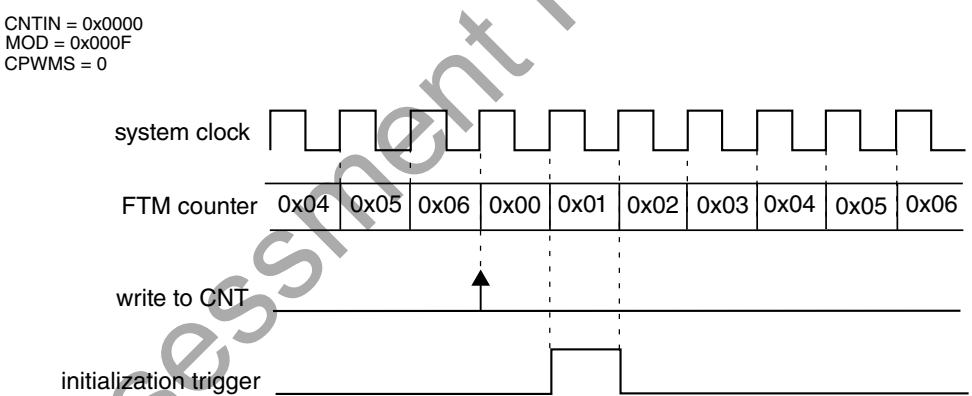


Figure 38-87. Initialization trigger is generated when there is a write to CNT register

For Assessment Purposes Only

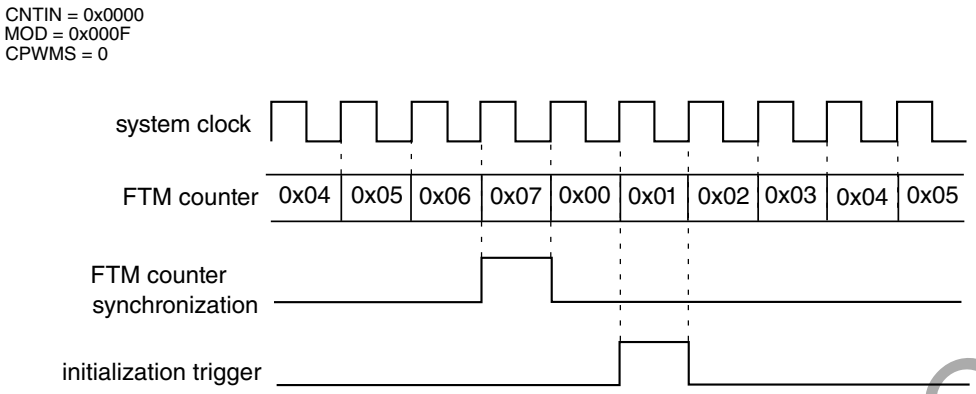


Figure 38-88. Initialization trigger is generated when there is the FTM counter synchronization

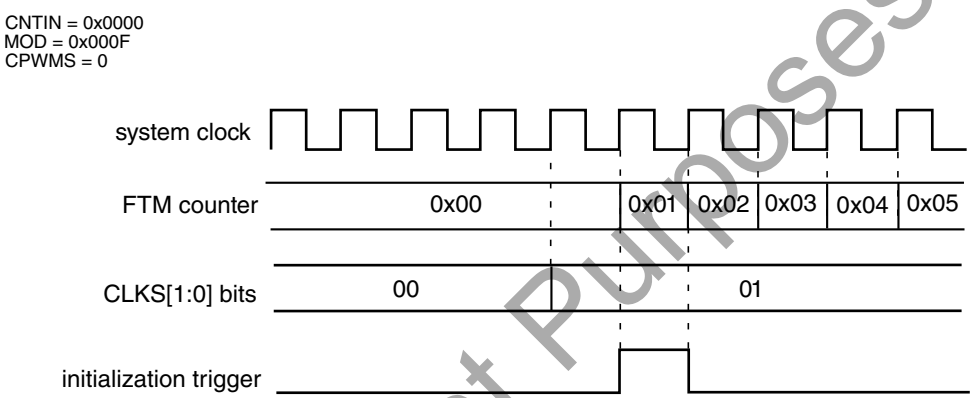
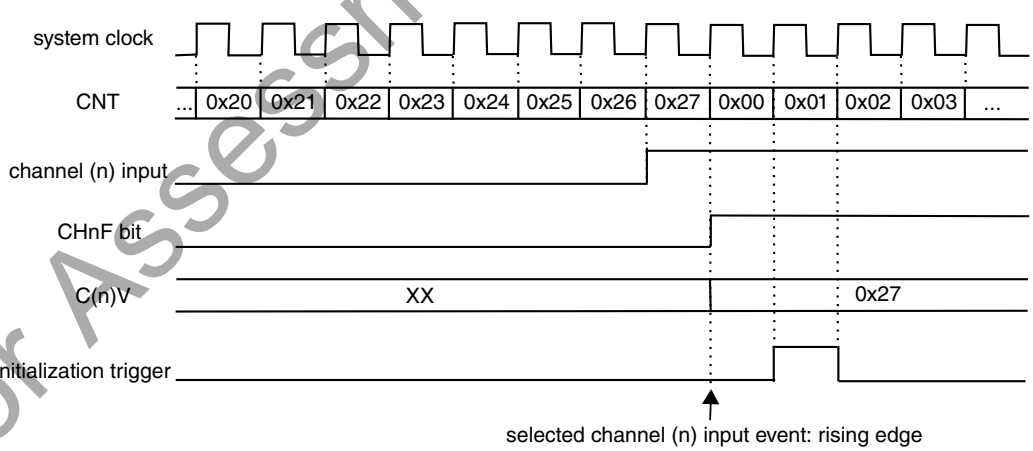


Figure 38-89. Initialization trigger is generated if (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits



NOTE
Channel (n) input after its synchronizer and filter
MOD = 0xFFFF
CNTIN = 0x0000
PS[2:0] = 3'b000
ICRST = 1'b1

Figure 38-90. Initialization trigger is generated if the channel (n) is in Input Capture mode, ICRST = 1 and the selected input capture event occurs in the channel (n) input

For Assessment Purposes Only

The initialization trigger output provides a trigger signal that is used for on-chip modules.

Note

- When FTM is in up-down count mode (CPWMS = 1), the initialization trigger can be generated according to loadpoints CNTMAX and CNTMIN at SYNC register if ITRIGR=1.

38.5.26 Capture Test Mode

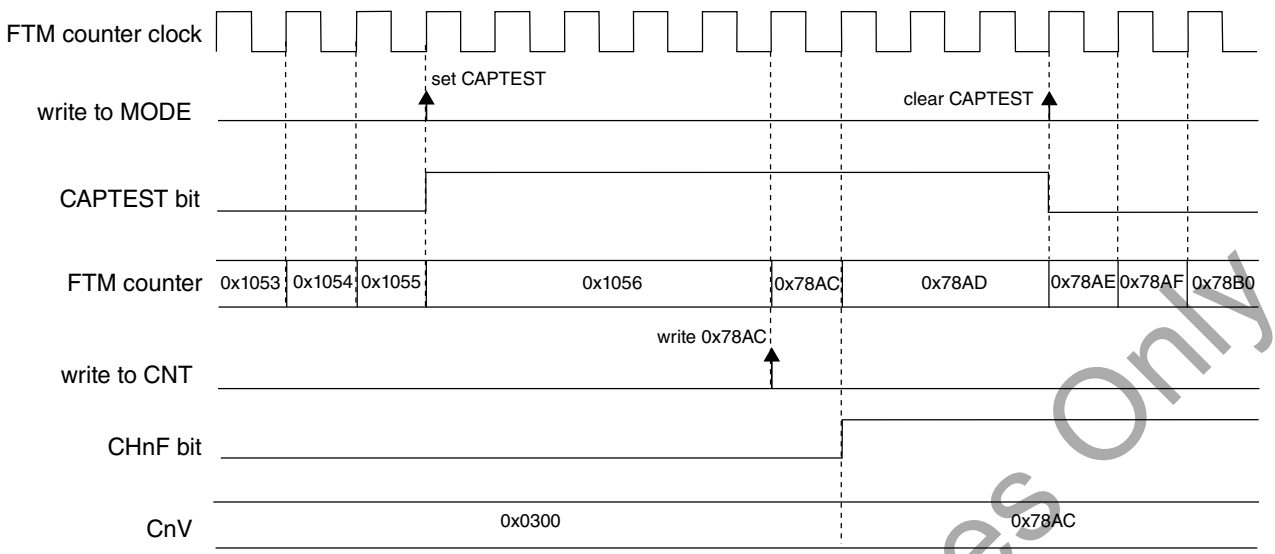
The Capture Test mode allows to test the CnV registers, the FTM counter and the interconnection logic between the FTM counter and CnV registers.

In this test mode, all channels must be configured for Input Capture mode and FTM counter must be configured to the Up counting.

When the Capture Test mode is enabled (CAPTEST = 1), the FTM counter is frozen and any write to CNT register updates directly the FTM counter; see the following figure. After it was written, all CnV registers are updated with the written value to CNT register and CHnF bits are set. Therefore, the FTM counter is updated with its next value according to its configuration. Its next value depends on CNTIN, MOD, and the written value to FTM counter.

The next reads of CnV registers return the written value to the FTM counter and the next reads of CNT register return FTM counter next value.

For Assessment Purposes Only



NOTE
- FTM counter is free running and (FTMEN = 1);
- FTM channel (n) is in Input Capture Mode.

Figure 38-91. Capture Test Mode

38.5.27 DMA

The channel generates a DMA transfer request according to DMA and CHnIE bits. See the following table.

Table 38-19. Channel DMA transfer request

DMA	CHnIE	Channel DMA Transfer Request	Channel Interrupt
0	0	The channel DMA transfer request is not generated.	The channel interrupt is not generated.
0	1	The channel DMA transfer request is not generated.	The channel interrupt is generated if (CHnF = 1).
1	0	The channel DMA transfer request is not generated.	The channel interrupt is not generated.
1	1	The channel DMA transfer request is generated if (CHnF = 1).	The channel interrupt is not generated.

If DMA = 1, the CHnF bit is cleared either by channel DMA transfer done or reading CnSC while CHnF is set and then writing a zero to CHnF bit according to CHnIE bit. See the following table.

For Development Purposes Only

Table 38-20. Clear CHnF bit when DMA = 1

CHnIE	How CHnF Bit Can Be Cleared
0	CHnF bit is cleared either when the channel DMA transfer is done or by reading CnSC while CHnF is set and then writing a 0 to CHnF bit.
1	CHnF bit is cleared when the channel DMA transfer is done.

38.5.28 Dual Edge Capture mode

The Dual Edge Capture mode is selected if DECAPEN = 1. This mode allows to measure a pulse width or period of the signal on the input of channel (n) of a channel pair. The channel (n) filter can be active in this mode when n is 0 or 2.

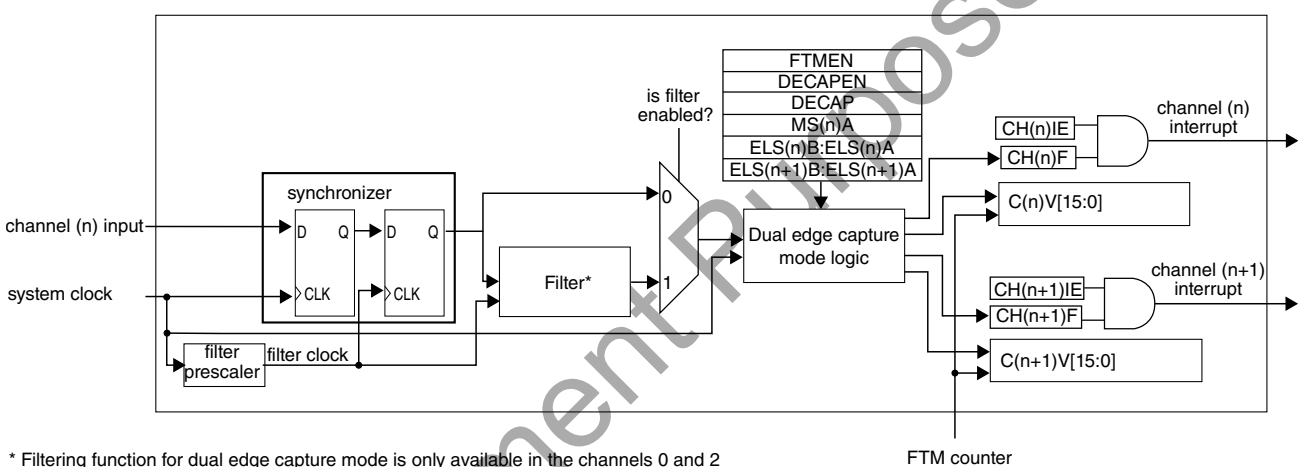


Figure 38-92. Dual Edge Capture mode block diagram

The MS(n)A bit defines if the Dual Edge Capture mode is one-shot or continuous.

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). If both ELS(n)B:ELS(n)A and ELS(n+1)B:ELS(n+1)A bits select the same edge, then it is the period measurement. If these bits select different edges, then it is a pulse width measurement.

In the Dual Edge Capture mode, only channel (n) input is used and channel (n+1) 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 and (CH(n)F = 1), then CH(n+1)F bit is set and the channel (n+1) interrupt is generated (if CH(n+1)IE = 1).

For Assessment Purposes Only

The C(n)V register stores the value of FTM counter when the selected edge by channel (n) is detected at channel (n) input. The C(n+1)V register stores the value of FTM counter when the selected edge by channel (n+1) is detected at channel (n) input.

In this mode, a coherency mechanism ensures coherent data when the C(n)V and C(n +1)V registers are read. The only requirement is that C(n)V must be read before C(n +1)V.

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 Dual Edge 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 and the FTM counter in Free running counter.

38.5.28.1 One-Shot Capture mode

The One-Shot Capture mode is selected when (DECAPEN = 1), and (MS(n)A = 0). In this capture mode, only one pair of edges at the channel (n) input is captured. The ELS(n)B:ELS(n)A bits select the first edge to be captured, and ELS(n+1)B:ELS(n+1)A bits select the second edge to be captured.

The edge captures are enabled while DECAP bit is set. For each new measurement in One-Shot Capture mode, first the CH(n)F and CH(n+1) bits must be cleared, and then the DECAP bit must be set.

In this mode, the DECAP bit is automatically cleared by FTM when the edge selected by channel (n+1) is captured. Therefore, while DECAP bit is set, the one-shot capture is in process. When this bit is cleared, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

Similarly, when the CH(n+1)F bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

For Assessment Purposes Only

38.5.28.2 Continuous Capture mode

The Continuous Capture mode is selected when (DECAPEN = 1), and (MS(n)A = 1). In this capture mode, the edges at the channel (n) input are captured continuously. The ELS(n)B:ELS(n)A bits select the initial edge to be captured, and ELS(n+1)B:ELS(n+1)A bits select the final edge to be captured.

The edge captures are enabled while DECAP bit is set. For the initial use, first the CH(n)F and CH(n+1)F bits must be cleared, and then DECAP bit must be set to start the continuous measurements.

When the CH(n+1)F bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers. The latest captured values are always available in these registers even after the DECAP bit is cleared.

In this mode, it is possible to clear only the CH(n+1)F bit. Therefore, when the CH(n+1)F bit is set again, the latest captured values are available in C(n)V and C(n+1)V registers.

For a new sequence of the measurements in the Dual Edge Capture – Continuous mode, clear the CH(n)F and CH(n+1)F bits to start new measurements.

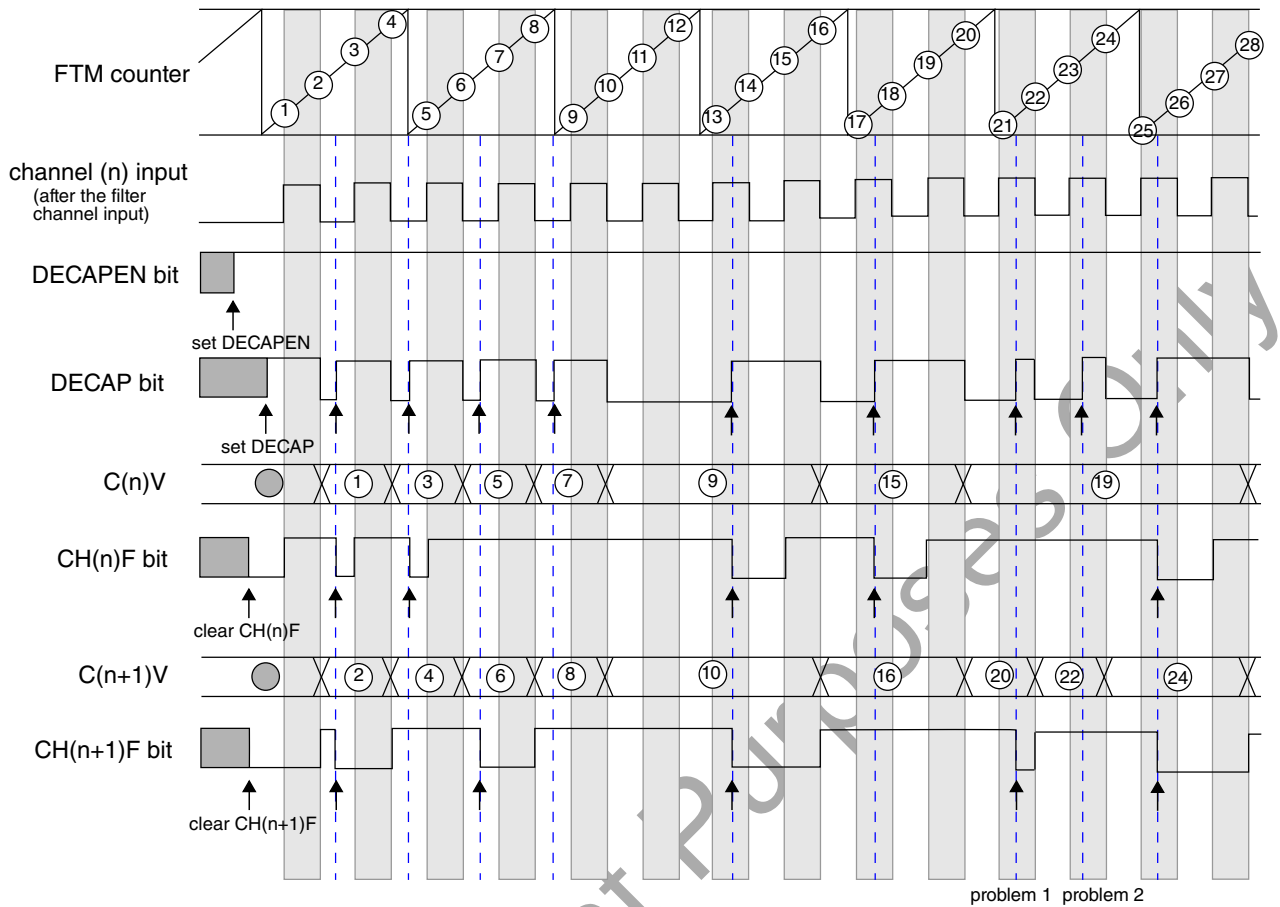
38.5.28.3 Pulse width measurement

If the channel (n) is configured to capture rising edges (ELS(n)B:ELS(n)A = 0:1) and the channel (n+1) to capture falling edges (ELS(n+1)B:ELS(n+1)A = 1:0), then the positive polarity pulse width is measured. If the channel (n) is configured to capture falling edges (ELS(n)B:ELS(n)A = 1:0) and the channel (n+1) to capture rising edges (ELS(n +1)B:ELS(n+1)A = 0:1), then the negative polarity pulse width is measured.

The pulse width measurement can be made in One-Shot Capture mode or Continuous Capture mode.

The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next positive polarity pulse width. The CH(n)F bit is set when the first edge of this pulse is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set and DECAP bit is cleared when the second edge of this pulse is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. Both DECAP and CH(n+1)F bits indicate when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.

For Assessment Purposes Only

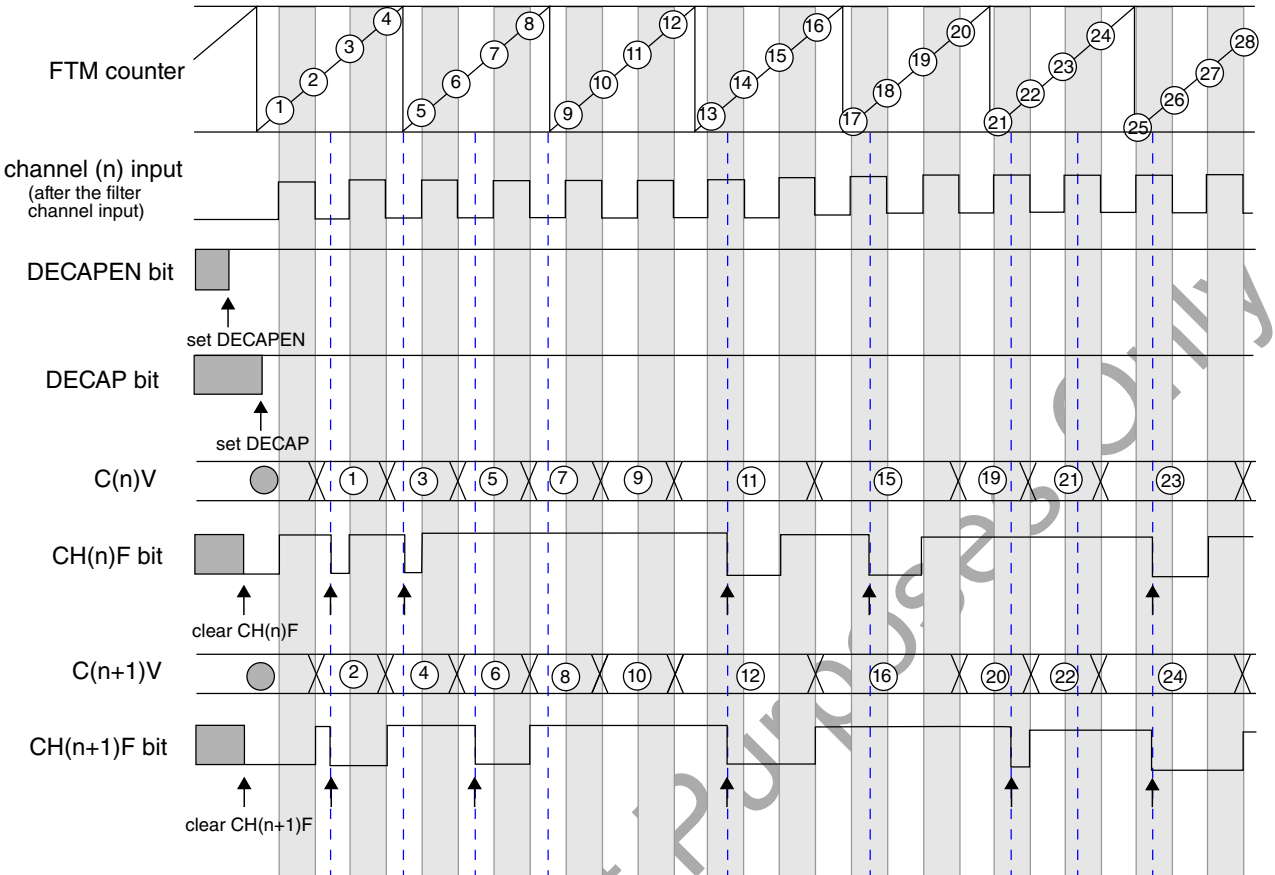


Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 38-93. Dual Edge Capture – One-Shot mode for positive polarity pulse width measurement

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The CH(n)F bit is set when the first edge of the positive polarity pulse is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set when the second edge of this pulse is detected, that is, the edge selected by ELS(n +1)B:ELS(n+1)A bits. The CH(n+1)F bit indicates when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.

For Assessment Purposes Only



Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.

Figure 38-94. Dual Edge Capture – Continuous mode for positive polarity pulse width measurement

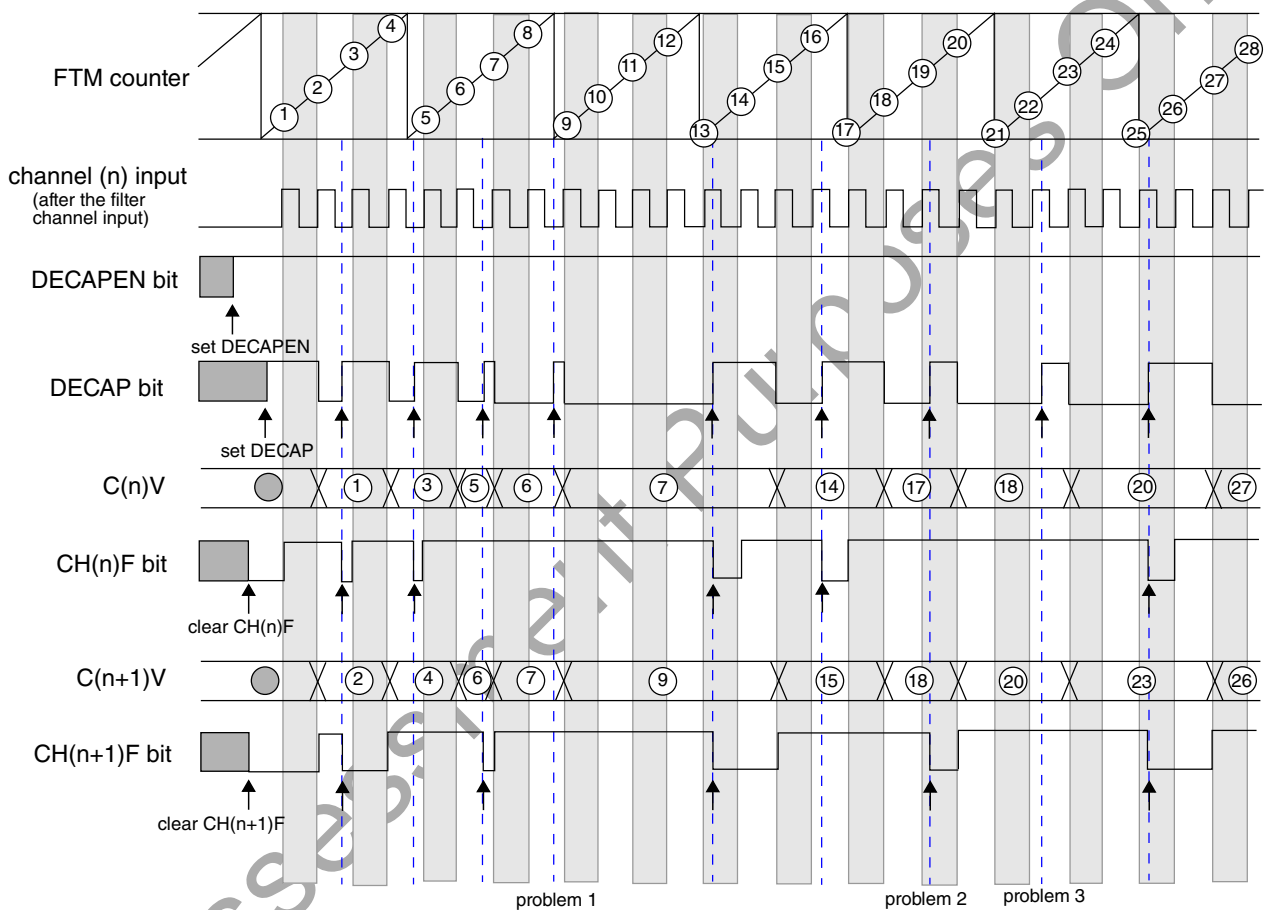
38.5.28.4 Period measurement

If the channels (n) and (n+1) are configured to capture consecutive edges of the same polarity, then the period of the channel (n) input signal is measured. If both channels (n) and (n+1) are configured to capture rising edges (ELS(n)B:ELS(n)A = 0:1 and ELS(n+1)B:ELS(n+1)A = 0:1), then the period between two consecutive rising edges is measured. If both channels (n) and (n+1) are configured to capture falling edges (ELS(n)B:ELS(n)A = 1:0 and ELS(n+1)B:ELS(n+1)A = 1:0), then the period between two consecutive falling edges is measured.

The period measurement can be made in One-Shot Capture mode or Continuous Capture mode.

For Assessment Purposes Only

The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next period. The CH(n)F bit is set when the first rising edge is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set and DECAP bit is cleared when the second rising edge is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. Both DECAP and CH(n+1)F bits indicate when two selected edges were captured and the C(n)V and C(n+1)V registers are ready for reading.



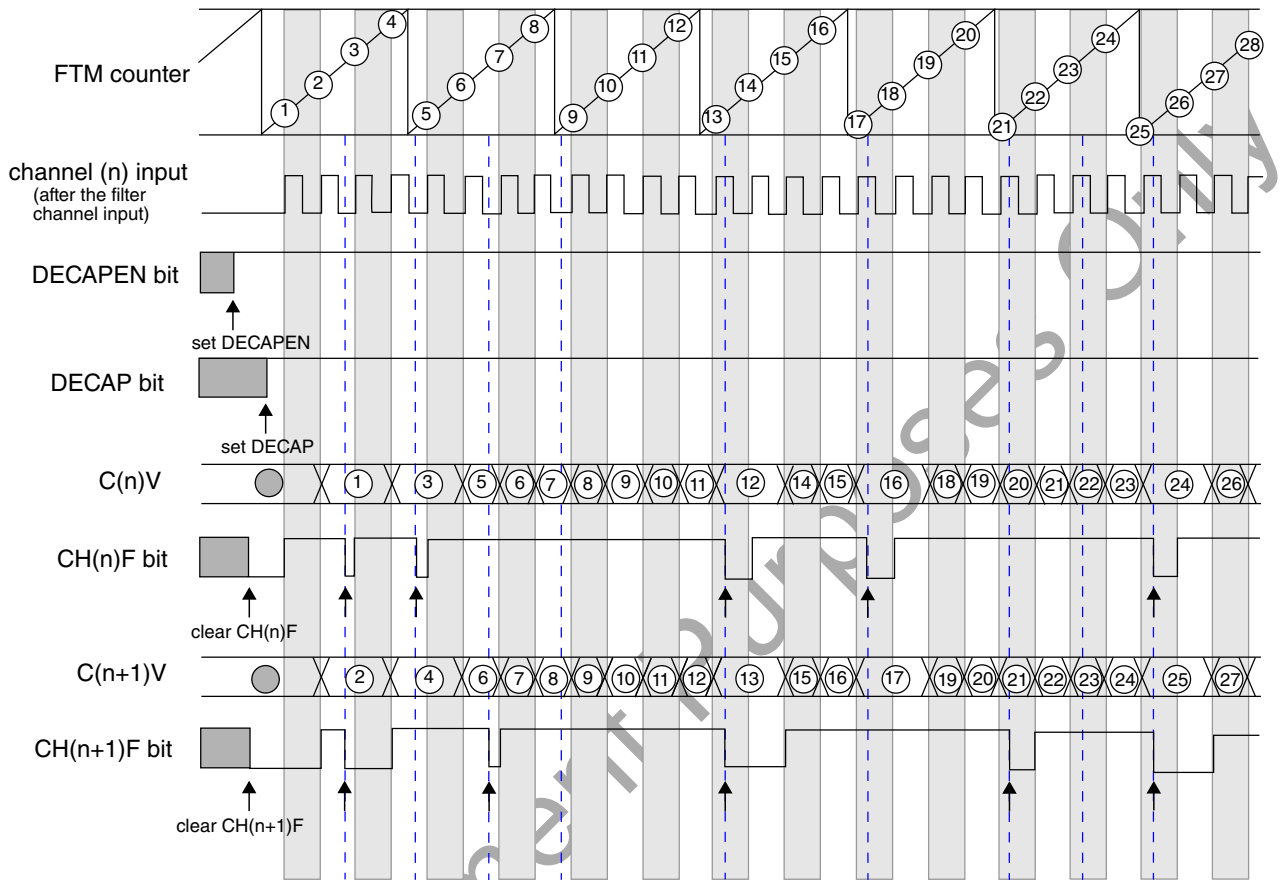
Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 0, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 3: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 38-95. Dual Edge Capture – One-Shot mode to measure of the period between two consecutive rising edges

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The CH(n)F bit is set when the first rising edge is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set

For Assessment Purposes Only

when the second rising edge is detected, that is, the edge selected by ELS(n+1)B:ELS(n +1)A bits. The CH(n+1)F bit indicates when two edges of the period were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.

Figure 38-96. Dual Edge Capture – Continuous mode to measure of the period between two consecutive rising edges

For Assessment Purposes Only

38.5.28.5 Read coherency mechanism

The Dual Edge Capture mode implements a read coherency mechanism between the FTM counter value captured in C(n)V and C(n+1)V registers. The read coherency mechanism is illustrated in the following figure. In this example, the channels (n) and (n +1) are in Dual Edge Capture – Continuous mode for positive polarity pulse width measurement. Thus, the channel (n) is configured to capture the FTM counter value when there is a rising edge at channel (n) input signal, and channel (n+1) to capture the FTM counter value when there is a falling edge at channel (n) input signal.

When a rising edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n) capture buffer. The channel (n) capture buffer value is transferred to C(n)V register when a falling edge occurs in the channel (n) input signal. C(n)V register has the FTM counter value when the previous rising edge occurred, and the channel (n) capture buffer has the FTM counter value when the last rising edge occurred.

When a falling edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n+1) capture buffer. The channel (n+1) capture buffer value is transferred to C(n+1)V register when the C(n)V register is read.

In the following figure, the read of C(n)V returns the FTM counter value when the event 1 occurred and the read of C(n+1)V returns the FTM counter value when the event 2 occurred.

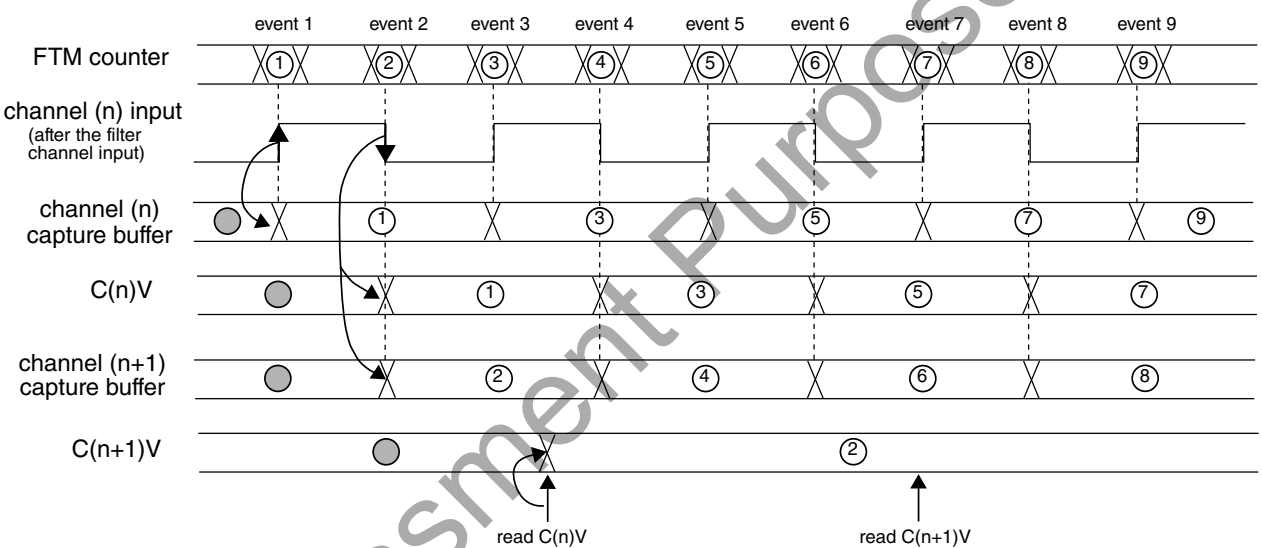


Figure 38-97. Dual Edge Capture mode read coherency mechanism

C(n)V register must be read prior to C(n+1)V register in dual edge capture one-shot and continuous modes for the read coherency mechanism works properly.

38.5.29 Quadrature Decoder mode

The Quadrature Decoder mode is selected if (QUADEN = 1). The Quadrature Decoder mode uses the input signals phase A and B to control the FTM counter increment and decrement. The following figure shows the quadrature decoder block diagram.

For Assessment Purposes Only

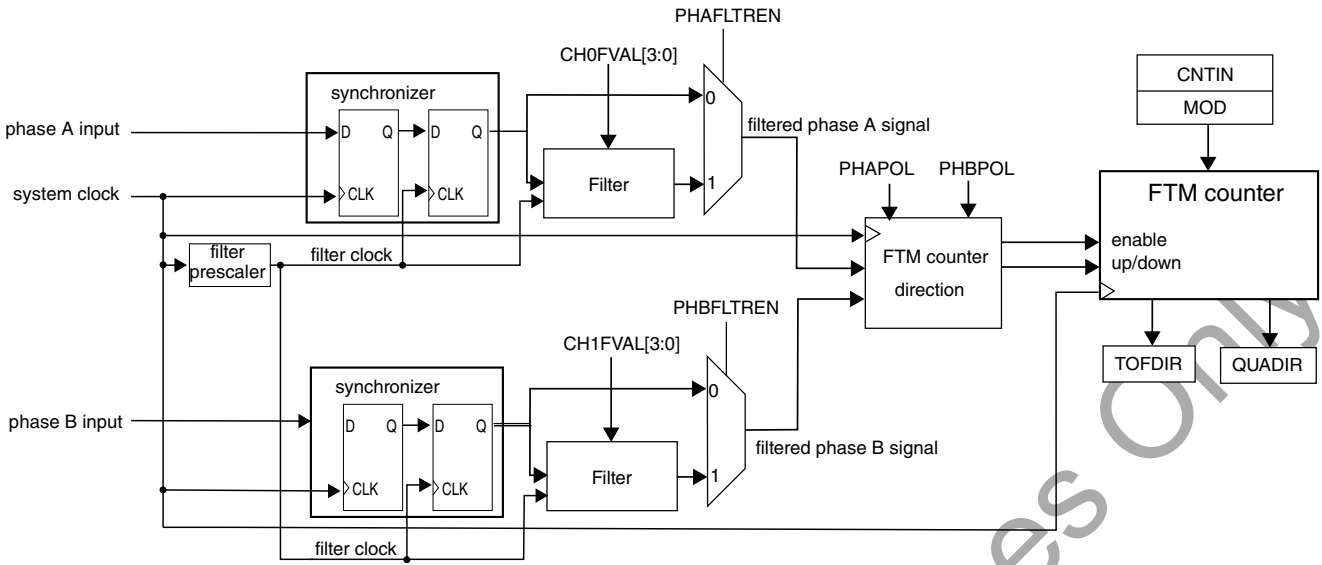


Figure 38-98. Quadrature Decoder block diagram

Each one of input signals phase A and B has a filter that is equivalent to the filter used in the channels input; Filter for Input Capture mode. The phase A input filter is enabled by PHAFLTREN bit and this filter's value is defined by CH0FVAL[3:0] bits (CH(n)FVAL[3:0] bits in FILTER0 register). The phase B input filter is enabled by PHBFLTREN bit and this filter's value is defined by CH1FVAL[3:0] bits (CH(n+1)FVAL[3:0] bits in FILTER0 register). The FLTPS[3:0] bits controls both filter prescalers.

Except for CH0FVAL[3:0] and CH1FVAL[3:0] bits, no channel logic is used in Quadrature Decoder mode.

Note

Notice that the FTM counter is clocked by the phase A and B input signals when quadrature decoder mode is selected. Therefore it is expected that the Quadrature Decoder be used only with the FTM channels in input capture or output compare modes.

Note

An edge at phase A must not occur together an edge at phase B and vice-versa.

The PHAPOL bit selects the polarity of the phase A input, and the PHBPOL bit selects the polarity of the phase B input.

For Assessment Purposes Only

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 phase B input value indicates the counting direction, and the phase A input defines the counting rate. The FTM counter is updated when there is a rising edge at phase A input signal.

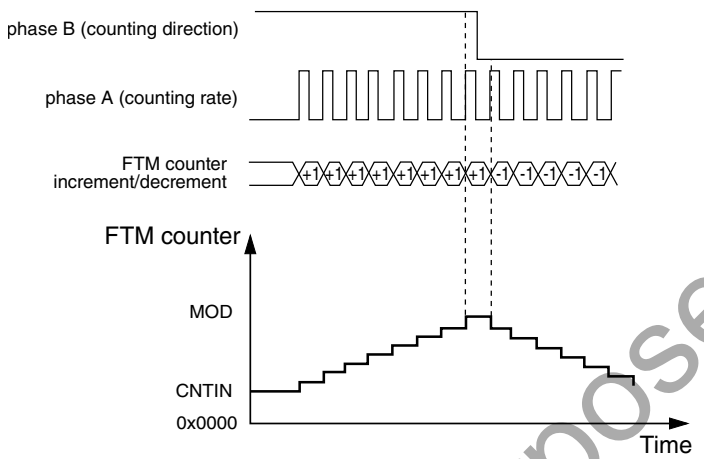


Figure 38-99. Quadrature Decoder – Count and Direction Encoding mode

If QUADMODE = 0, then the Phase A and Phase B Encoding mode is enabled; see the following figure. In this mode, the relationship between phase A and B signals indicates the counting direction, and phase A and B signals define the counting rate. The FTM counter is updated when there is an edge either at the phase A or phase B signals.

If PHAPOL = 0 and PHBPOL = 0, then the FTM counter increment happens when:

- there is a rising edge at phase A signal and phase B signal is at logic zero;
- there is a rising edge at phase B signal and phase A signal is at logic one;
- there is a falling edge at phase B signal and phase A signal is at logic zero;
- there is a falling edge at phase A signal and phase B signal is at logic one;

and the FTM counter decrement happens when:

- there is a falling edge at phase A signal and phase B signal is at logic zero;
- there is a falling edge at phase B signal and phase A signal is at logic one;
- there is a rising edge at phase B signal and phase A signal is at logic zero;
- there is a rising edge at phase A signal and phase B signal is at logic one.

For Assessment Purposes Only

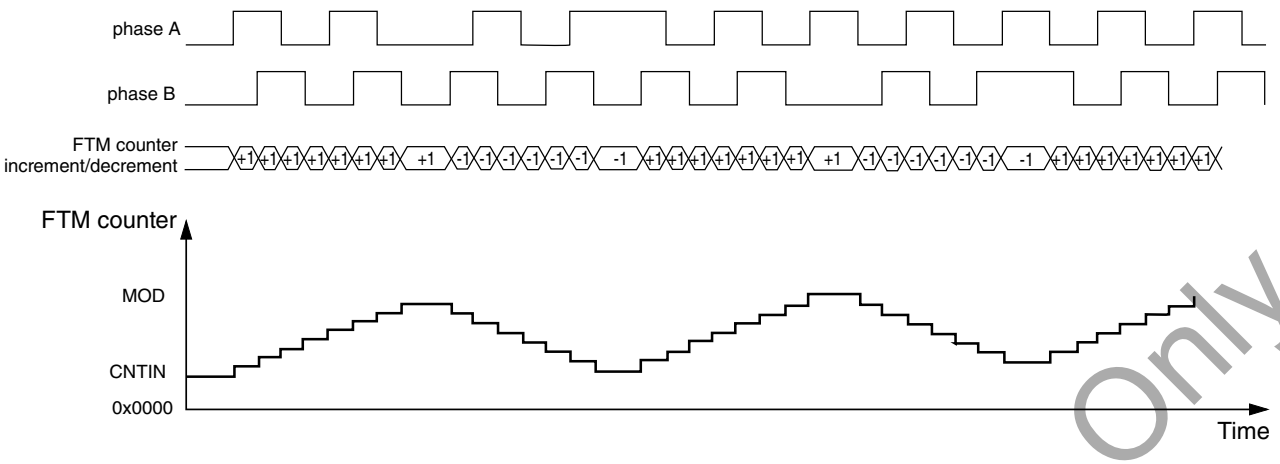


Figure 38-100. Quadrature Decoder – Phase A and Phase B Encoding mode

The following figure shows the FTM counter overflow in up counting. In this case, when the FTM counter changes from MOD to CNTIN, TOF and TOFDIR bits are set. TOF bit indicates the FTM counter overflow occurred. TOFDIR indicates the counting was up when the FTM counter overflow occurred.

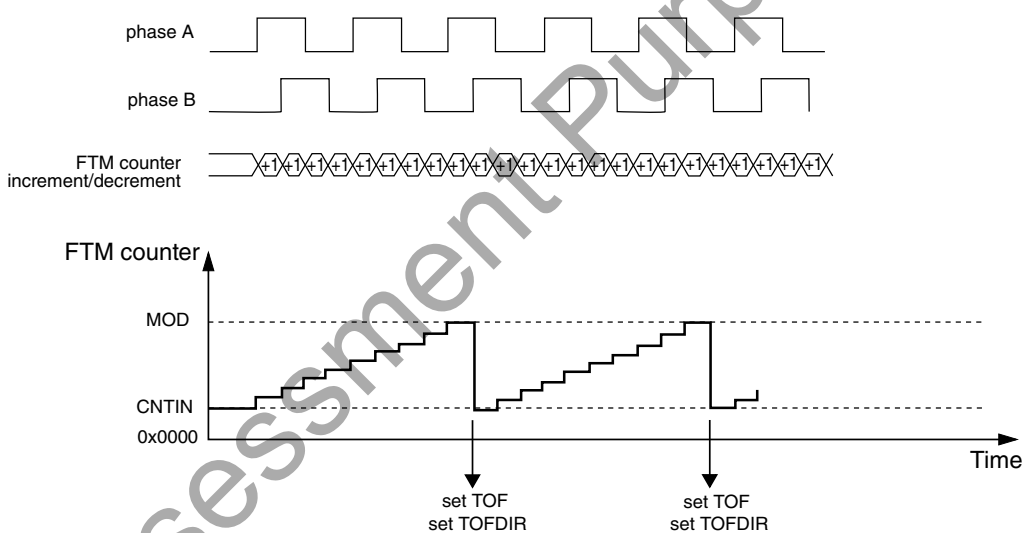


Figure 38-101. FTM Counter overflow in up counting for Quadrature Decoder mode

The following figure shows the FTM counter overflow in down counting. In this case, when the FTM counter changes from CNTIN to MOD, TOF bit is set and TOFDIR bit is cleared. TOF bit indicates the FTM counter overflow occurred. TOFDIR indicates the counting was down when the FTM counter overflow occurred.

For Assessment Purposes Only

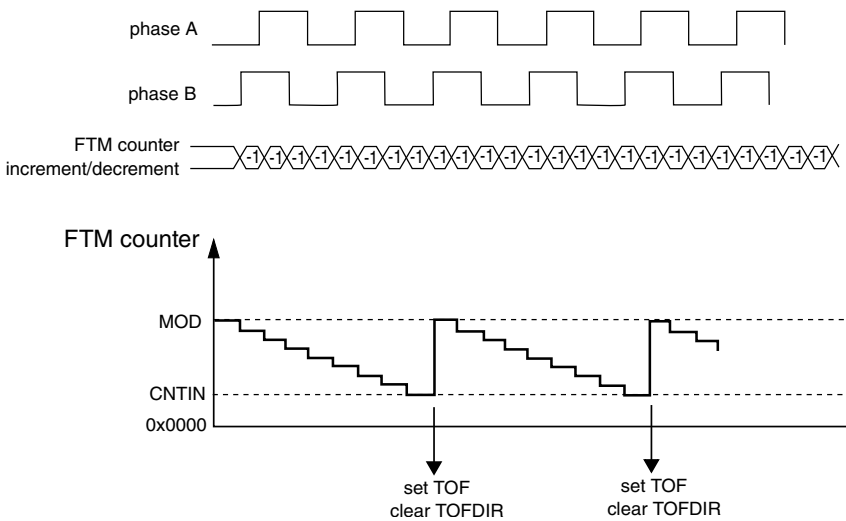


Figure 38-102. FTM counter overflow in down counting for Quadrature Decoder mode

38.5.29.1 Quadrature Decoder boundary conditions

The following figures show the FTM counter responding to motor jittering typical in motor position control applications.

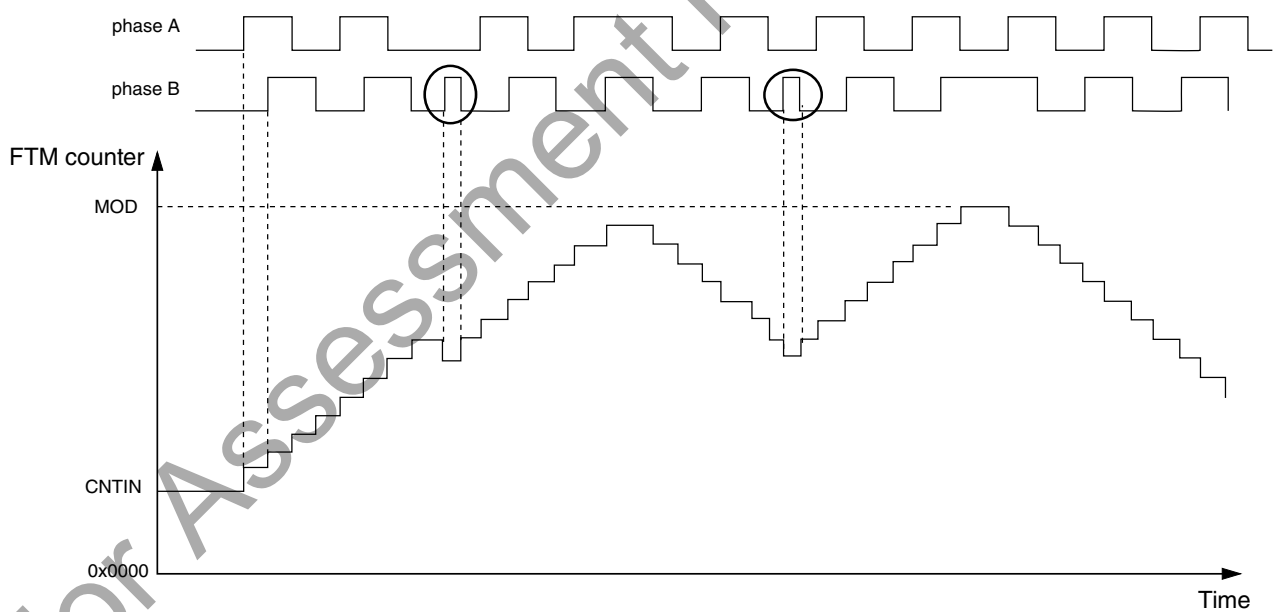


Figure 38-103. Motor position jittering in a mid count value

The following figure shows motor jittering produced by the phase B and A pulses respectively:

For Assessment Purposes Only

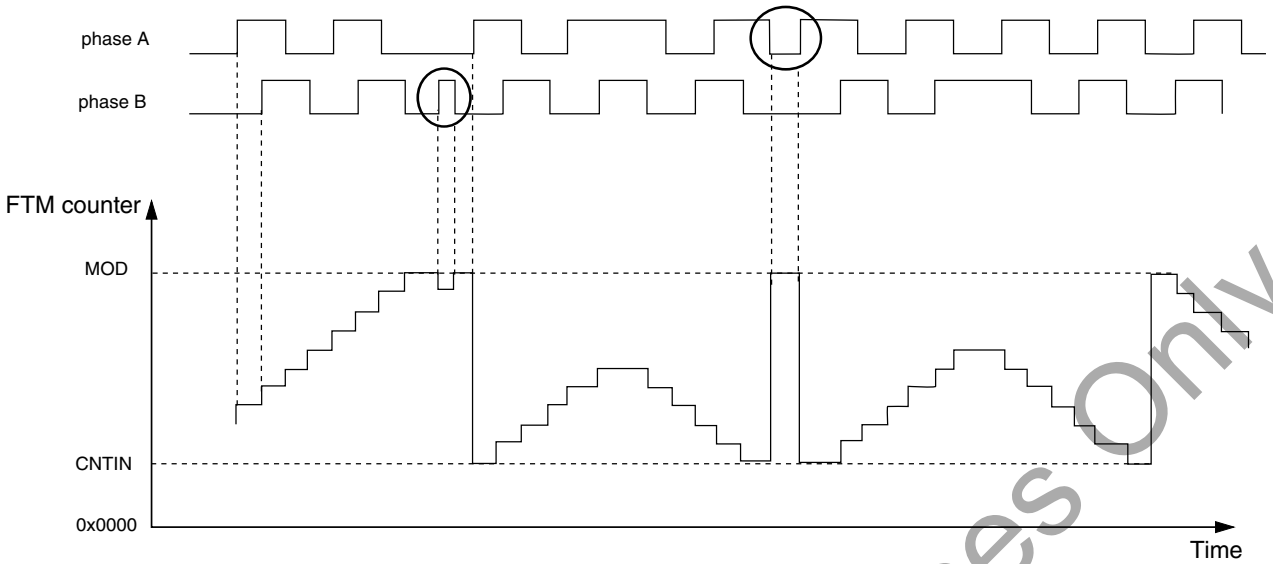


Figure 38-104. Motor position jittering near maximum and minimum count value

The first highlighted transition causes a jitter on the FTM counter value near the maximum count value (MOD). The second indicated transition occurs on phase A and causes the FTM counter transition between the maximum and minimum count values which are defined by MOD and CNTIN registers.

The appropriate settings of the phase A and phase B input filters are important to avoid glitches that may cause oscillation on the FTM counter value. The preceding figures show examples of oscillations that can be caused by poor input filter setup. Thus, it is important to guarantee a minimum pulse width to avoid these oscillations.

38.5.30 Debug mode

When the chip is in Debug mode, the BDMMODE[1:0] bits select the behavior of the FTM counter, the CH(n)F bit, the channels output, and the writes to the MOD, CNTIN, and C(n)V registers according to the following table.

Table 38-21. FTM behavior when the chip Is in Debug mode

BDMMODE	FTM Counter	CH(n)F Bit	FTM Channels Output	Writes to MOD, CNTIN, and C(n)V Registers
00	Stopped	can be set	Functional mode	Writes to these registers bypass the registers buffers
01	Stopped	is not set	The channels outputs are forced to their safe value according to POLn bit	Writes to these registers bypass the registers buffers
10	Stopped	is not set	The channels outputs are frozen when the chip enters in Debug mode	Writes to these registers bypass the registers buffers

Table continues on the next page...

Assessment Purposes Only

Table 38-21. FTM behavior when the chip Is in Debug mode (continued)

BDMMODE	FTM Counter	CH(n)F Bit	FTM Channels Output	Writes to MOD, CNTIN, and C(n)V Registers
11	Functional mode	can be set	Functional mode	Functional mode

Note that if BDMMODE[1:0] = 2’b00 then the channels outputs remain at the value when the chip enters in Debug mode, because the FTM counter is stopped. However, the following situations modify the channels outputs in this Debug mode.

- Write any value to CNT register; see Counter reset. In this case, the FTM counter is updated with the CNTIN register value and the channels outputs are updated to the initial value – except for those channels set to Output Compare mode.
- FTM counter is reset by PWM Synchronization mode; see FTM counter synchronization. In this case, the FTM counter is updated with the CNTIN register value and the channels outputs are updated to the initial value – except for channels in Output Compare mode.
- In the channels outputs initialization, the channel (n) output is forced to the CH(n)OI bit value when the value 1 is written to INIT bit. See Initialization.

Note

The BDMMODE[1:0] = 2’b00 must not be used with the Fault control. Even if the fault control is enabled and a fault condition exists, the channels outputs values are updated as above.

Note

If CLKS[1:0] = 2'b00 in BDM, a non-zero value is written to CLKS in BDM, and CnV = CNTIN when the BDM is disabled, then the CHnF bit is set (since if the channel is a 0% EPWM signal) when the BDM is disabled.

38.5.31 Global Load

The global load mechanism allows several modules to have their double buffered registers synchronously reloaded after a synchronization event if a write to one operation is performed in the global load OK (GLDOK) bit in the FTM_PWMLOAD register. Global load may be enabled or disabled configuring the global load enable (GLEN) bit in

For Assessment Purposes Only

the FTM_PWMLOAD register. Writing one in the GLDOK bit with GLEN enabled has the same effect of writing one in the LDOK bit. Refer to SoC specific information about global load connections.

Global load mechanism allows MOD, HCR, CNTIN, and C(n)V registers to be updated with the content of the register buffer at configurable reload point. The figure below shows an example of connection between FTM global load inputs and outputs considering that GLDOK bit is implemented outside from FTM module.

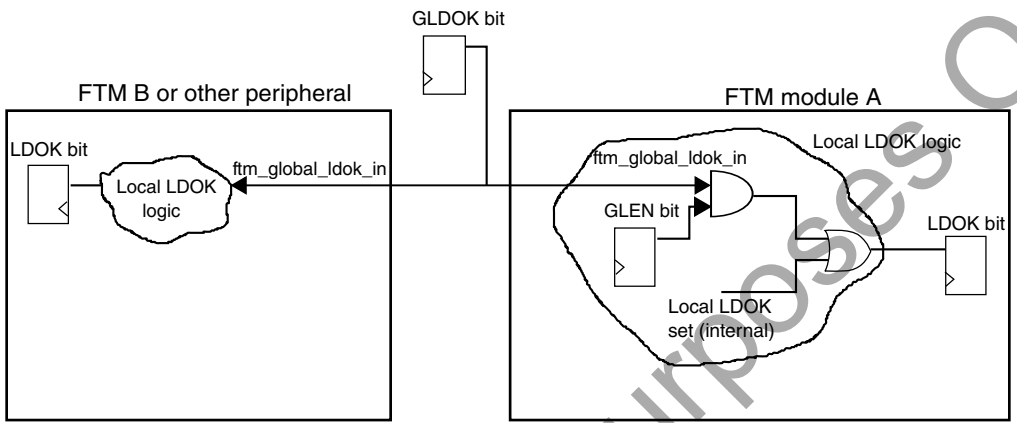


Figure 38-105. Global load logic

38.5.32 Global time base (GTB)

The global time base (GTB) is a FTM function that allows the synchronization of multiple FTM modules on a chip. The following figure shows an example of the GTB feature used to synchronize two FTM modules. In this case, the FTM A and B channels can behave as if just one FTM module was used, that is, a global time base.

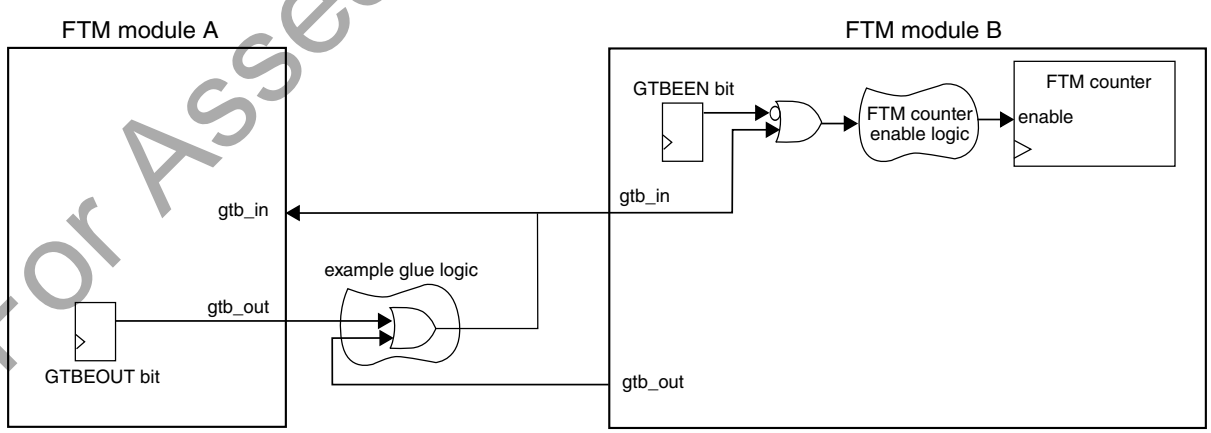


Figure 38-106. Global time base (GTB) block diagram

For Assessment Purposes Only

The GTB functionality is implemented by the GTBEEN and GTBEOUT bits in the CONF register, the input signal *gtb_in*, and the output signal *gtb_out*. The GTBEEN bit enables gtb_in to control the FTM counter enable signal:

- If GTBEEN = 0, each one of FTM modules works independently according to their configured mode.
- If GTBEEN = 1, the FTM counter update is enabled only when gtb_in is 1.

In the configuration described in the preceding figure, FTM modules A and B have their FTM counters enabled if at least one of the gtb_out signals from one of the FTM modules is 1. There are several possible configurations for the interconnection of the gtb_in and gtb_out signals, represented by the example glue logic shown in the figure. Note that these configurations are chip-dependent and implemented outside of the FTM modules. See the chip-specific FTM information for the chip's specific implementation.

NOTE

- In order to use the GTB signals to synchronize the FTM counter of different FTM modules, the configuration of each FTM module should guarantee that its FTM counter starts counting as soon as the gtb_in signal is 1.
- The GTB feature does not provide continuous synchronization of FTM counters, meaning that the FTM counters may lose synchronization during FTM operation. The GTB feature only allows the FTM counters to *start* their operation synchronously.

38.5.32.1 Enabling the global time base (GTB)

To enable the GTB feature, follow these steps for each participating FTM module:

1. Stop the FTM counter: Write 00b to SC[CLKS].
2. Program the FTM to the intended configuration. The FTM counter mode needs to be consistent across all participating modules.
3. Write 1 to CONF[GTBEEN] and write 0 to CONF[GTBEOUT] at the same time.
4. Select the intended FTM counter clock source in SC[CLKS]. The clock source needs to be consistent across all participating modules.
5. Reset the FTM counter: Write any value to the CNT register.

To initiate the GTB feature in the configuration described in the preceding figure, write 1 to CONF[GTBEOUT] in the FTM module used as the time base.

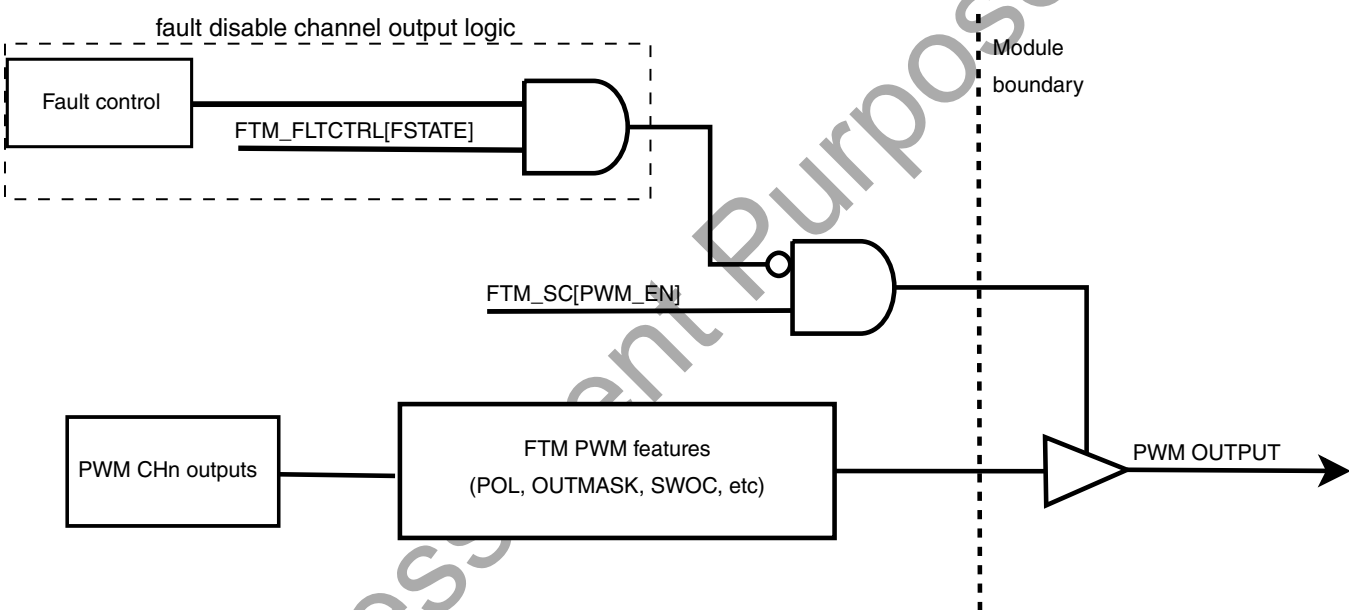
For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

38.5.33 Output Logic

The following figure shows the output logic of each FTM channel output including how each PWM output has individual fault disabling and output enable. This allows flexibility regarding the external circuitry interface.

The output buffer logic depends on PWM_EN bit of FTM_SC register and fault disable channel output logic (see Fault control to more details) . Channel outputs will be enabled only if PWM_EN is enabled and there is no fault event ongoing configured to tri-state the outputs by FSTATE bit at FTM_FLTCTRL register. Note that Polarity logic will act before channel enable logic. Therefore, it is imperative that the user program the channel polarities before enabling the output pins. A fault condition can result in the PWM output being tristated, forced to a logic 1, or forced to a logic 0 depending on the values programmed into the POL, and FSTATE fields.



38.6 Reset overview

The FTM is reset whenever any chip reset occurs.

When the FTM exits from reset:

- the FTM counter and the prescaler counter are zero and are stopped (CLKS[1:0] = 00b);
- the timer overflow interrupt is zero, see Timer Overflow Interrupt;
- the channels interrupts are zero, see Channel (n) Interrupt;
- the fault interrupt is zero, see Fault Interrupt;
- the channels are in input capture mode, see Input Capture mode;

For Assessment Purposes Only

- the channels outputs are zero;
- the channels pins are not controlled by FTM (ELS(n)B:ELS(n)A = 0:0) (See the table in the description of CnSC register).

The following figure shows the FTM behavior after the reset. At the reset (item 1), the FTM counter is disabled (see the description of the CLKS field in the Status and Control register), its value is updated to zero and the pins are not controlled by FTM (See the table in the description of CnSC register).

After the reset, the FTM should be configurated (item 2). It is necessary to define the FTM counter mode, the FTM counting limits (MOD and CNTIN registers value), the channels mode and CnV registers value according to the channels mode.

Thus, it is recommended to write any value to CNT register (item 3). This write updates the FTM counter with the CNTIN register value and the channels output with its initial value (except for channels in output compare mode) (Counter reset).

The next step is to select the FTM counter clock by the CLKS[1:0] bits (item 4). It is important to highlight that the pins are only controlled by FTM when CLKS[1:0] bits are different from zero (See the table in the description of CnSC register).

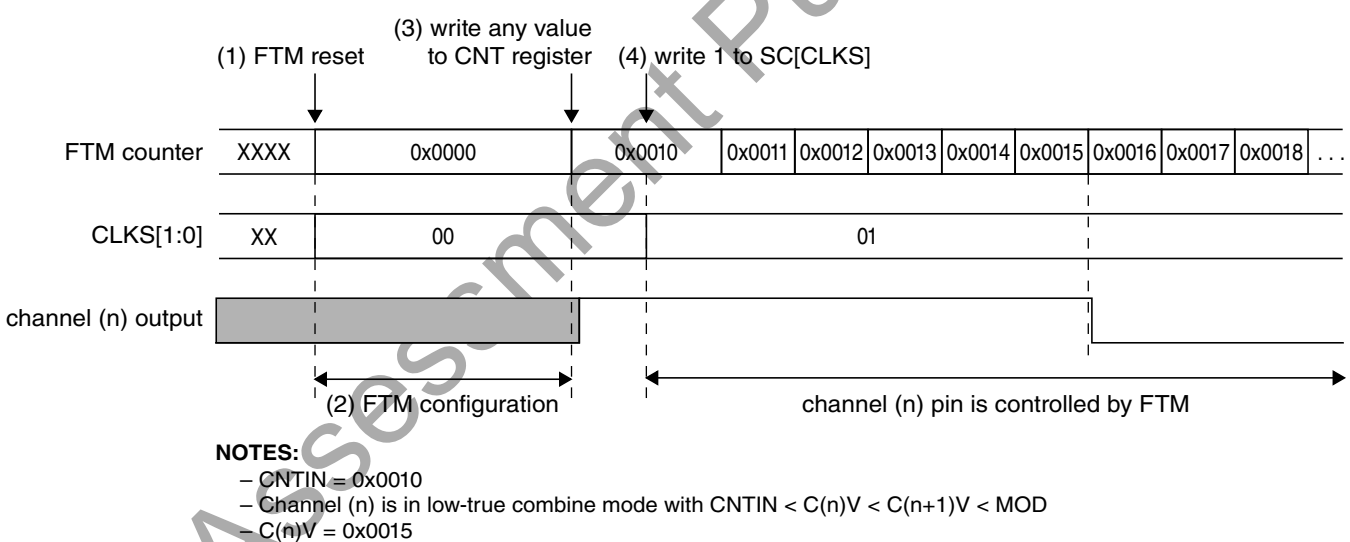
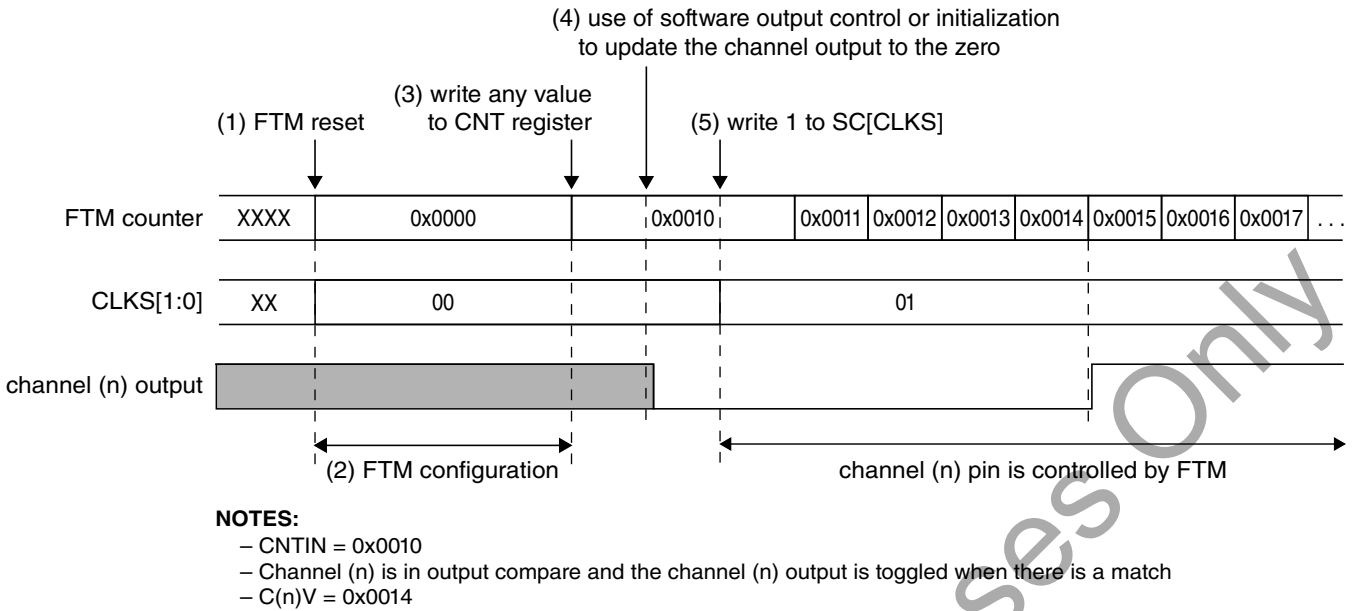


Figure 38-107. FTM behavior after reset when the channel (n) is in Combine mode

The following figure shows an example when the channel (n) is in Output Compare mode and the channel (n) output is toggled when there is a match. In the Output Compare mode, the channel output is not updated to its initial value when there is a write to CNT register (item 3). In this case, use the software output control (Software Output Control Mode) or the initialization (Initialization) to update the channel output to the selected value (item 4).

For Assessment Purposes Only



NOTES:
- CNTIN = 0x0010
- Channel (n) is in output compare and the channel (n) output is toggled when there is a match
- C(n)V = 0x0014

Figure 38-108. FTM behavior after reset when the channel (n) is in Output Compare mode

38.7 FTM Interrupts

38.7.1 Timer Overflow Interrupt

The timer overflow interrupt is generated when (TOIE = 1) and (TOF = 1).

38.7.2 Reload Point Interrupt

The Reload Point interrupt is generated when (RIE = 1) and (RF = 1).

38.7.3 Channel (n) Interrupt

The channel (n) interrupt is generated when (CHnIE = 1) and (CHnF = 1).

38.7.4 Fault Interrupt

The fault interrupt is generated when (FAULTIE = 1) and (FAULTF = 1).

For Assessment Purposes Only

38.8 Initialization Procedure

The following initialization procedure is recommended to configure the FlexTimer operation. This procedure can also be used to do a new configuration of the FlexTimer operation.

1. Define the POL bits.
2. Mask the channels outputs using SYNCHOM = 0. Two clocks after the write to OUTMASK, the channels output are in the safe value.
3. (Re)Configuration FTM counter and channels to generation of periodic signals:
   - Disable the clock. If the selected mode is Quadrature Decoder, then disable this mode.
   - Write to MOD.
   - Write to CNTIN.
   - Configure the channels that will be used.
   - Select the high-true and low-true channels modes.
   - Write to CnV for all channels that will be used .
   - (Re)Configure deadtime and fault control.
   - Do not use the SWOC without SW synchronization (see item 6).
   - Do not use the Inverting without SW synchronization (see item 6).
   - Do not use the Initialization.
   - Do not change the polarity control.
   - Do not configure the HW synchronization
4. Write any value to CNT. The FTM Counter is reset and the channels output are updated according to new configuration.
5. Enable the clock. Write to CLKS[1:0] bits a value different from zero. If in the Quadrature Decoder mode, enable this mode.
6. Configure the SW synchronization for SWOC (if it is necessary), Inverting (if it is necessary) and Output Mask (always)
   - Select synchronization for Output Mask Write to SYNC (SWSYNC = 0, TRIG2 = 0, TRIG1 = 0, TRIG0 = 0, SYNCHOM = 1, REINIT = 0, CNTMAX = 0, CNTMIN = 0)
   - Write to SYNCONF.
     - HW Synchronization can not be enabled (HWSOC = 0, HWINVC = 0, HWOM = 0, HWWRBUF = 0, HWRSTCNT = 0, HWTRIGMODE = 0).
     - SW Synchronization for SWOC (if it is necessary): SWSOC = [0/1] and SWOC = [0/1].
     - SW Synchronization for Inverting (if it is necessary): SWINVC = [0/1] and INVC = [0/1].

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- SW Synchronization for SWOM (always): SWOM = 1. No enable the SW Synchronization for write buffers (because the writes to registers with write buffer are done using CLKS[1:0] = 2'b00): SWWRBUF = 0 and CNTINC = 0 .
- SW Synchronization for counter reset (always): SWRSTCNT = 1.
- Enhanced synchronization (always): SYNCMODE = 1.

- If the SWOC is used (SWSOC = 1 and SWOC = 1), then write to SWOCTRL register.
- If the Inverting is used (SWINVC = 1 and INVC = 1), then write to INVCTRL register.
- Write to OUTMASK to enable the masked channels.

7. Generate the Software Trigger Write to SYNC (SWSYNC = 1, TRIG2 = 0, TRIG1 = 0, TRIG0 = 0, SYNCHOM = 1, REINIT = 0, CNTMAX = 0, CNTMIN = 0)
8. Write to PWM_EN to enable the PWM outputs.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 39
Low Power Interrupt Timer (LPIT)

39.1 Chip-specific Low Power Interrupt Timer (LPIT) information

39.1.1 Instantiation Information

This device contains one LPIT module with four channels.

NOTE

Low leakage mode is not supported in this device.

39.1.1.1 LPIT/DMA Periodic Trigger Assignments

The LPIT generates periodic trigger events to the DMA Mux as shown in the table below.

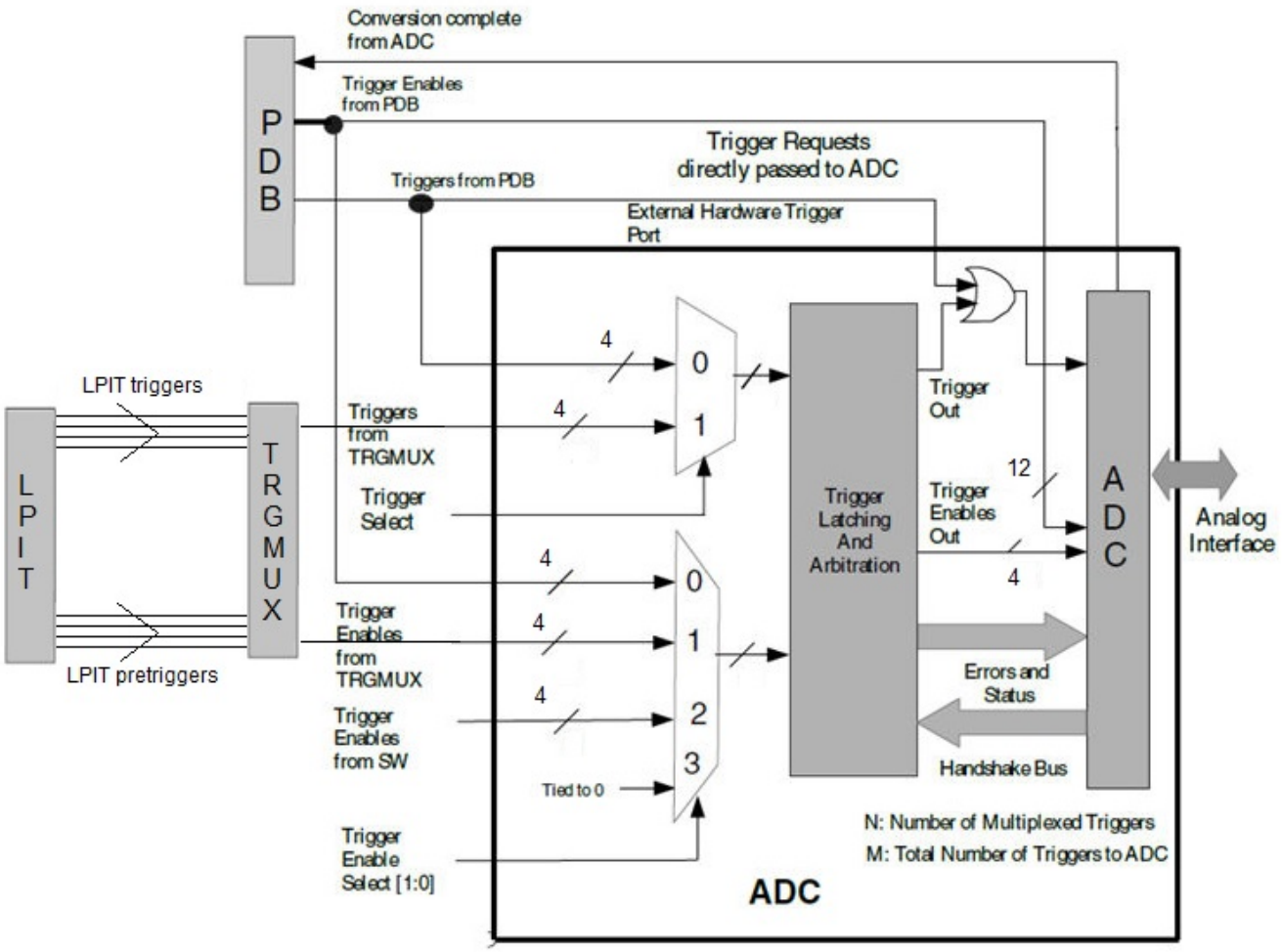
Table 39-1. LPIT channel assignments for periodic DMA triggering

DMA Channel Number	LPIT Channel
0	0
1	1
2	2
3	3

39.1.2 LPIT/ADC Trigger

The LPIT could be used as an alternate ADC hardware trigger source, the implementation is through TRGMUX. Each LPIT channel supports one pre-trigger and one trigger. The LPIT channels are implemented based on independent counters. When used as ADC trigger source, the channel outputs generate the ADC hardware trigger. The following diagram shows an example of using LPIT triggering ADC0.

For Assessment Purposes Only



39.2 Introduction

39.2.1 Overview

The Low Power Periodic Interrupt Timer (LPIT) is a multi-channel timer module generating independent pre-trigger and trigger outputs. These timer channels can operate individually or can be chained together. The LPIT can operate in low power modes if configured to do so. The pre-trigger and trigger outputs can be used to trigger other modules on the device.

Each timer channel can be configured to run independently and made to work in either compare or capture modes. In compare mode, the timers decrement when enabled and generate an output pre-trigger and timeout pulse. The trigger output is 1 clock cycle delayed of the pre-trigger pulse. Each timer channel start, reload and restart can be

For Assessm

controlled via control bits. The timer can be configured to always decrement, or decrement on selected trigger inputs or previous channel timeout (when channels are chained). By chaining timer channels, applications can achieve larger timeout durations. In capture mode, the timer can be used to perform measurements as the timer value is captured (in the timer value register) when a selected trigger input is asserted. In capture mode, the timer can support once-off or multiple measurements (for example, frequency measurements).

The timer channels operate on an asynchronous clock, which is independent from the register read/write access clock. Clock synchronization between the clock domains ensures normal operations.

39.2.2 Block Diagram

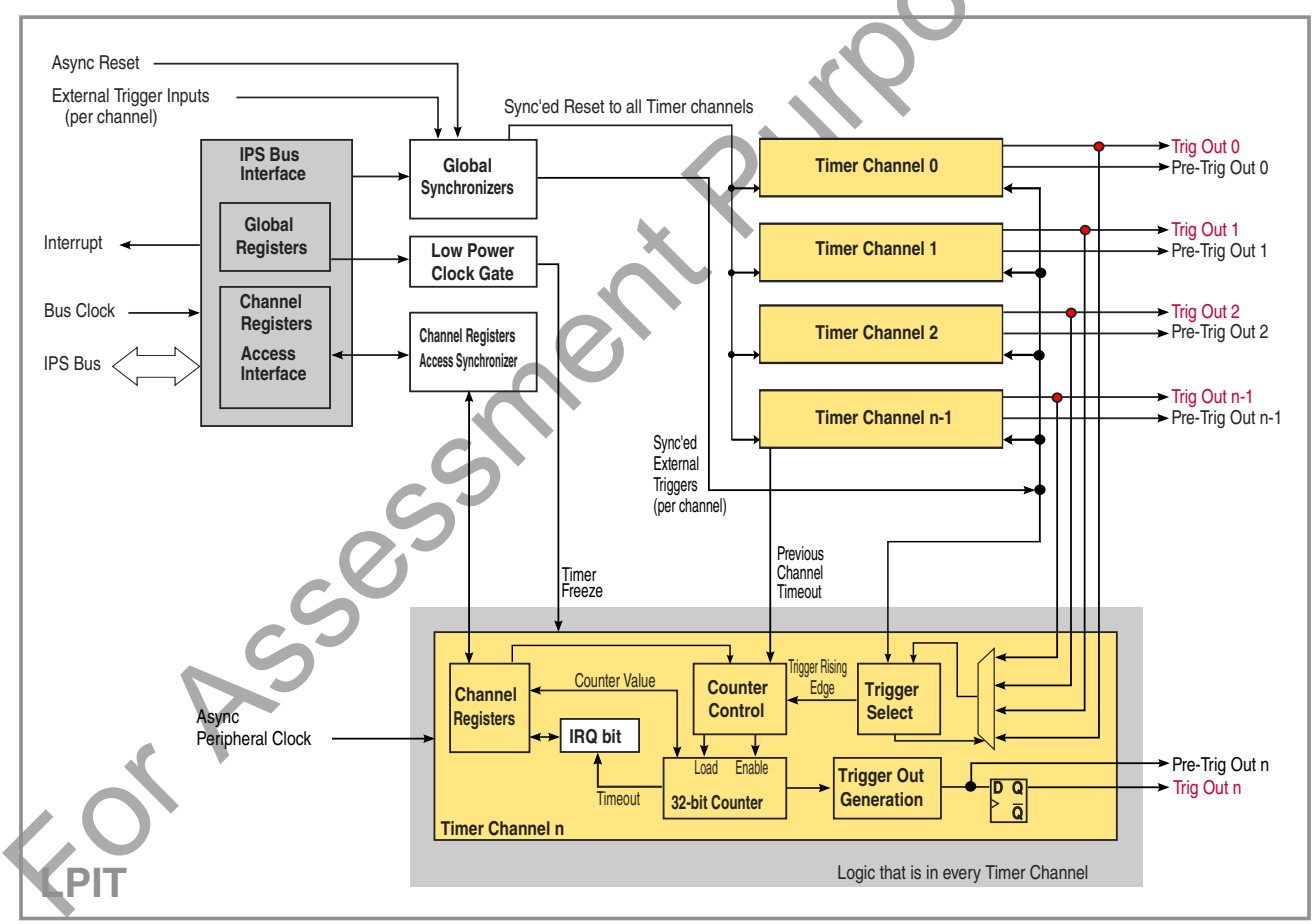


Figure 39-1. Top Level Block Diagram

For Assessment Purposes Only

39.3 Modes of operation

The LPIT module supports the chip modes described in the following table.

Table 39-2. Chip modes supported by the LPIT module

Chip mode	LPIT Operation
Run	Normal operation
Stop/Wait	Can continue operating provided the Doze Enable bit (MCR[DOZE_EN]) is clear and the LPIT is using an external or internal clock source which remains operating during stop/wait modes.
Low Leakage Stop	The Doze Enable (MCR[DOZE_EN]) bit is ignored and the LPIT will be disabled for the duration of low leakage mode.
Debug	Can continue operating provided the Debug Enable bit (MCR[DBG_EN]) is set.

39.4 Memory Map and Registers

39.4.1 LPIT Register Descriptions

The memory map comprises of 32-bit aligned registers which can be accessed via 8-bit, 16-bit, or 32-bit accesses. Write access to reserved locations will generate a transfer error. Read access to reserved locations will also generate a transfer error and the read data bus will show all 0s. The Memory Map and complete module is in Big Endian format.

The module will not check for correctness of programmed values in the registers and software must ensure that correct values are being written.

39.4.1.1 LPIT Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID Register (VERID)	32	RO	01000000h
4h	Parameter Register (PARAM)	32	RO	00000404h
8h	Module Control Register (MCR)	32	RW	00000000h
Ch	Module Status Register (MSR)	32	W1C	00000000h
10h	Module Interrupt Enable Register (MIER)	32	RW	00000000h

Table continues on the next page...

r Assessment Purpos

Offset	Register	Width (In bits)	Access	Reset value
14h	Set Timer Enable Register (SETTEN)	32	RW	00000000h
18h	Clear Timer Enable Register (CLRTEN)	32	WORZ	00000000h
20h	Timer Value Register (TVAL0)	32	RW	00000000h
24h	Current Timer Value (CVAL0)	32	RO	FFFFFFFFh
28h	Timer Control Register (TCTRL0)	32	RW	00000000h
30h	Timer Value Register (TVAL1)	32	RW	00000000h
34h	Current Timer Value (CVAL1)	32	RO	FFFFFFFFh
38h	Timer Control Register (TCTRL1)	32	RW	00000000h
40h	Timer Value Register (TVAL2)	32	RW	00000000h
44h	Current Timer Value (CVAL2)	32	RO	FFFFFFFFh
48h	Timer Control Register (TCTRL2)	32	RW	00000000h
50h	Timer Value Register (TVAL3)	32	RW	00000000h
54h	Current Timer Value (CVAL3)	32	RO	FFFFFFFFh
58h	Timer Control Register (TCTRL3)	32	RW	00000000h

39.4.1.2 Version ID Register (VERID)

39.4.1.2.1 Address

Register	Offset
VERID	0h

39.4.1.2.2 Diagram

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

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

39.4.1.2.3 Fields

Field	Function
0-15 FEATURE	Feature Number This read only field returns the feature set number.
16-23 MINOR	Minor Version Number This read only field returns the minor version number for the module specification
24-31 MAJOR	Major Version Number This read only field returns the major version number for the module specification

39.4.1.3 Parameter Register (PARAM)

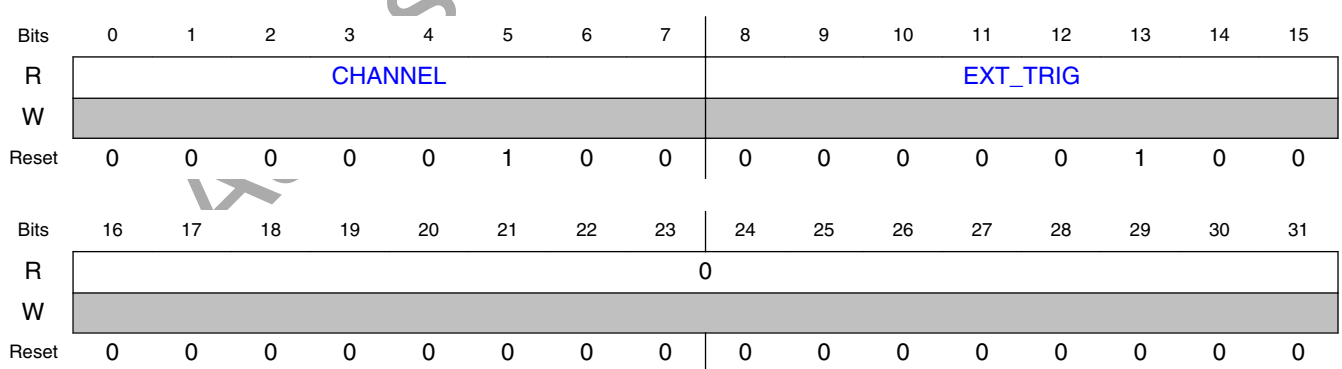
39.4.1.3.1 Address

Register	Offset
PARAM	4h

39.4.1.3.2 Function

This register provides details on the parameter settings that were used while including this module in the device.

39.4.1.3.3 Diagram



Freescale Confidential Proprietary - Non-Disclosure Agreement required

39.4.1.3.4 Fields

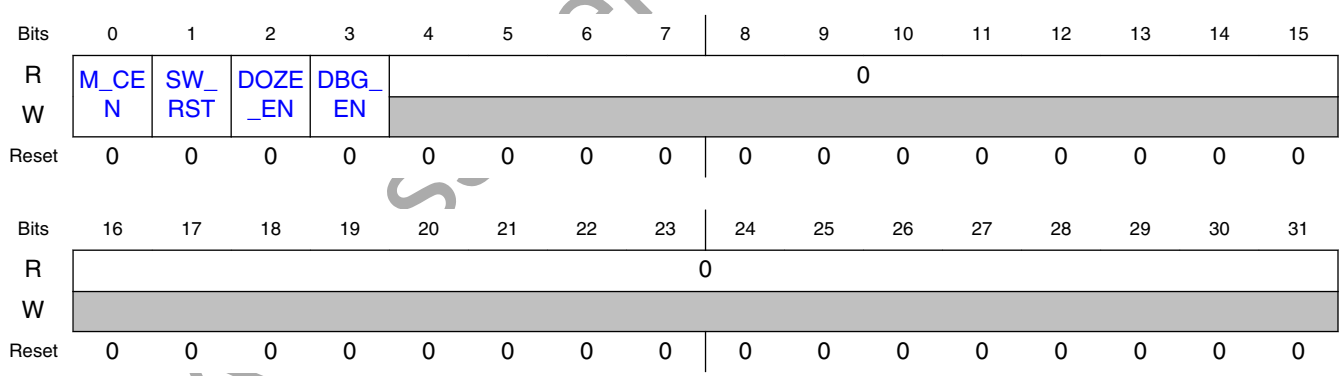
Field	Function
0-7 CHANNEL	Number of Timer Channels Number of timer channels implemented.
8-15 EXT_TRIG	Number of External Trigger Inputs Number of external triggers implemented.
16-31 —	This read-only field is reserved and always has the value 0

39.4.1.4 Module Control Register (MCR)

39.4.1.4.1 Address

Register	Offset
MCR	8h

39.4.1.4.2 Diagram



39.4.1.4.3 Fields

Field	Function
0 M_CEN	Module Clock Enable Enables the peripheral clock to the module timers. This bit must be asserted when writing to MSR, SETTEN , CLRTEN, TCTRL, and TVAL registers. Both clocks (bus clock and peripheral clock) must be enabled to allow for clock synchronization and update of register bits.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	NOTE: Writing to the MSR, SETTEN , CLRTEN, TCTRL, and TVAL registers while M_CEN = 0, will lead to assertion of transfer error for that bus cycle. Writing to CVAL and Reserved registers will always generate transfer error. **NOTE:** There might be additional clock gating bits in the device that stop the peripheral clock to this module. Please ensure those are configured appropraitely to enable the peripheral clock to this module. 0b - Peripheral clock to timers is disabled 1b - Peripheral clock to timers is enabled
1 SW_RST	Software Reset Bit Resets all channels and registers, except the Module Control Register. Remains set until cleared by software. 0b - Timer channels and registers are not reset 1b - Timer channels and registers are reset
2 DOZE_EN	DOZE Mode Enable Bit Allows the timer channels to be stopped or continue to run when the device enters the DOZE mode 0b - Timer channels are stopped in DOZE mode 1b - Timer channels continue to run in DOZE mode
3 DBG_EN	Debug Enable Bit Allows the timer channels to be stopped when the device enters the Debug mode 0b - Timer channels are stopped in Debug mode 1b - Timer channels continue to run in Debug mode
4-31 Reserved	This read-only field is reserved and always has the value 0

39.4.1.5 Module Status Register (MSR)

39.4.1.5.1 Address

Register	Offset
MSR	Ch

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

39.4.1.5.2 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TIF0	TIF1	TIF2	TIF3	0											
W	w1c	w1c	w1c	w1c												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

39.4.1.5.3 Fields

Field	Function
0 TIF0	Channel 0 Timer Interrupt Flag In compare modes, sets to 1 at the end of the timer period. In capture modes, sets to 1 when the trigger asserts. Writing logic 1 to this flag clears it. Writing 0 has no effect. 0b - Timer has not timed out 1b - Timeout has occurred
1 TIF1	Channel 1 Timer Interrupt Flag In compare modes, sets to 1 at the end of the timer period. In capture modes, sets to 1 when the trigger asserts. Writing logic 1 to this flag clears it. Writing 0 has no effect. 0b - Timer has not timed out 1b - Timeout has occurred
2 TIF2	Channel 2 Timer Interrupt Flag In compare modes, sets to 1 at the end of the timer period. In capture modes, sets to 1 when the trigger asserts. Writing logic 1 to this flag clears it. Writing 0 has no effect. 0b - Timer has not timed out 1b - Timeout has occurred
3 TIF3	Channel 3 Timer Interrupt Flag In compare modes, sets to 1 at the end of the timer period. In capture modes, sets to 1 when the trigger asserts. Writing logic 1 to this flag clears it. Writing 0 has no effect. 0b - Timer has not timed out 1b - Timeout has occurred
4-31 Reserved	This read-only field is reserved and always has the value 0.

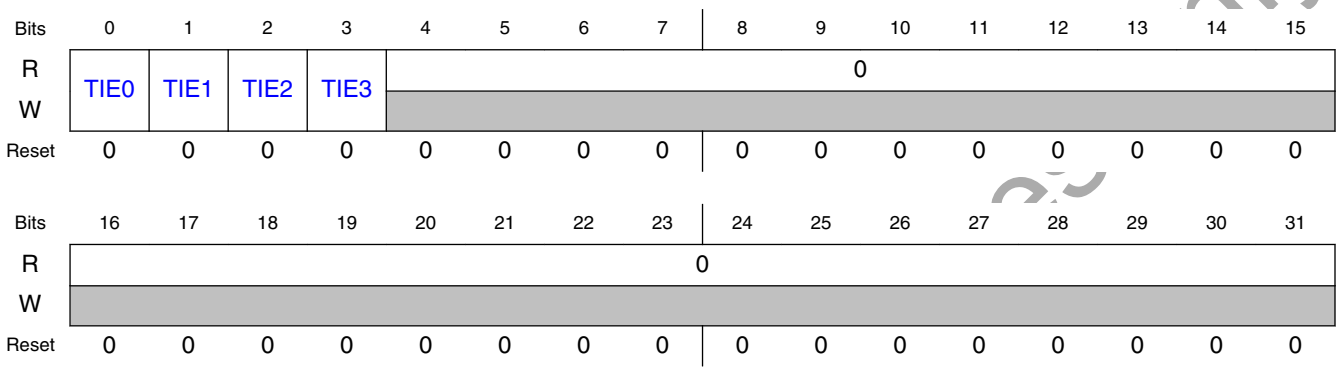
39.4.1.6 Module Interrupt Enable Register (MIER)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

39.4.1.6.1 Address

Register	Offset
MIER	10h

39.4.1.6.2 Diagram



39.4.1.6.3 Fields

Field	Function
0 TIE0	Channel 0 Timer Interrupt Enable Enables interrupt generation when this bit is set to 1 and if corresponding Timer Interrupt Flag is asserted. 0b - Interrupt generation is disabled 1b - Interrupt generation is enabled
1 TIE1	Channel 1 Timer Interrupt Enable Enables interrupt generation when this bit is set to 1 and if corresponding Timer Interrupt Flag is asserted. 0b - Interrupt generation is disabled 1b - Interrupt generation is enabled
2 TIE2	Channel 2 Timer Interrupt Enable Enables interrupt generation when this bit is set to 1 and if corresponding Timer Interrupt Flag is asserted. 0b - Interrupt generation is disabled 1b - Interrupt generation is enabled
3 TIE3	Channel 3 Timer Interrupt Enable Enables interrupt generation when this bit is set to 1 and if corresponding Timer Interrupt Flag is asserted. 0b - Interrupt generation is disabled 1b - Interrupt generation is enabled
4-31 Reserved	This read-only field is reserved and always has the value 0.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

39.4.1.7 Set Timer Enable Register (SETTEN)

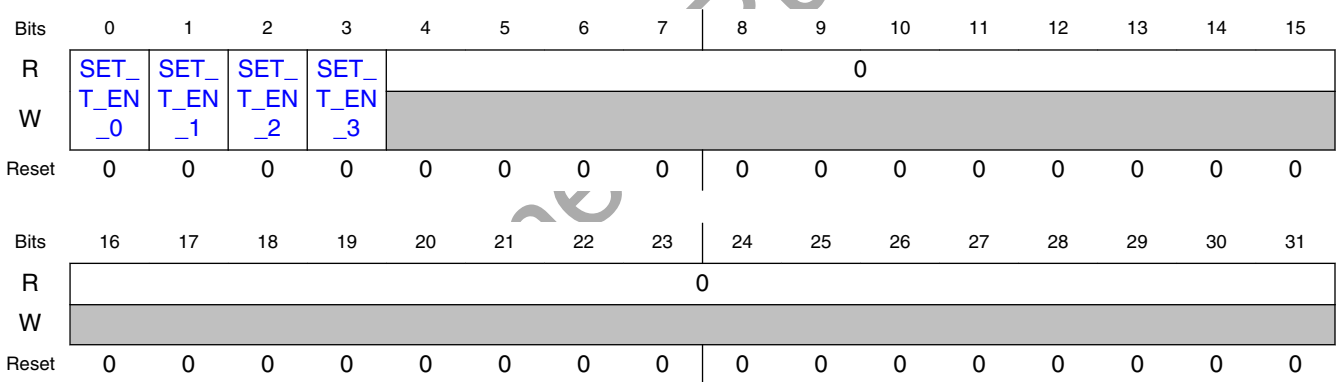
39.4.1.7.1 Address

Register	Offset
SETTEN	14h

39.4.1.7.2 Function

This register allows simulataneous enabling of timer channels. Timer channels can be enabled either by writing '1' to T_EN in respective TCTRLn register or setting the corresponding bit in this register. Writing a '0' to this register has no effect. CLRTEN register should be used to disable timer channels simultaneously.

39.4.1.7.3 Diagram



39.4.1.7.4 Fields

Field	Function
0 SET_T_EN_0	Set Timer 0 Enable Writing '1' to this bit will enable the timer channel 0. This bit can be used in addition to T_EN bit in TCTRL0 register. Writing a 0 will not disable the counter. This bit will be cleared when T_EN bit in TCTRL0 is set to 0 or '1' is written to the CLR_T_EN_0 bit in CLRTEN register. 0b - No effect 1b - Enables the Timer Channel 0
1 SET_T_EN_1	Set Timer 1 Enable Writing '1' to this bit will enable the timer channel 1. This bit can be used in addition to T_EN bit in TCTRL1 register. Writing a 0 will not disable the counter. This bit will be cleared when T_EN bit in TCTRL1 is set to '0' or '1' is written to the CLR_T_EN_1 bit in CLRTEN register.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0b - No Effect 1b - Enables the Timer Channel 1
2 SET_T_EN_2	Set Timer 2 Enable Writing '1' to this bit will enable the timer channel 2. This bit can be used in addition to T_EN bit in TCTRL2 register. Writing a 0 will not disable the counter. This bit will be cleared when T_EN bit in TCTRL2 is set to '0' or '1' is written to the CLR_T_EN_2 bit in CLRTEN register. 0b - No Effect 1b - Enables the Timer Channel 2
3 SET_T_EN_3	Set Timer 3 Enable Writing '1' to this bit will enable the timer channel 3. This bit can be used in addition to T_EN bit in TCTRL3 register. Writing a 0 will not disable the counter. This bit will be cleared when T_EN bit in TCTRL3 is set to '0' or '1' is written to the CLR_T_EN_3 bit in CLRTEN register. 0b - No effect 1b - Enables the Timer Channel 3
4-31 Reserved	This read-only field is reserved and always has the value 0.

39.4.1.8 Clear Timer Enable Register (CLRTEN)

39.4.1.8.1 Address

Register	Offset
CLRTEN	18h

39.4.1.8.2 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0	0	0											
W	CLR_T_EN_0	CLR_T_EN_1	CLR_T_EN_2	CLR_T_EN_3												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

39.4.1.8.3 Fields

Field	Function
0 CLR_T_EN_0	Clear Timer 0 Enable Writing a '1' to this bit will disable the timer channel 0. This bit can be used in addition to T_EN bit in TCTRL0 register. Writing a 1 will not enable the counter. This bit is self clearing and will always read 0. 0b - No action 1b - Clear T_EN bit for Timer Channel 0
1 CLR_T_EN_1	Clear Timer 1 Enable Writing a '1' to this bit will disable the timer channel 1. This bit can be used in addition to T_EN bit in TCTRL1 register. Writing a 1 will not enable the counter. This bit is self clearing and will always read 0. 0b - No Action 1b - Clear T_EN bit for Timer Channel 1
2 CLR_T_EN_2	Clear Timer 2 Enable Writing a '1' to this bit will disable the timer channel 2. This bit can be used in addition to T_EN bit in TCTRL2 register. Writing a 1 will not enable the counter. This bit is self clearing and will always read 0. 0b - No Action 1b - Clear T_EN bit for Timer Channel 2
3 CLR_T_EN_3	Clear Timer 3 Enable Writing a '1' to this bit will disable the timer channel 3. This bit can be used in addition to T_EN bit in TCTRL3 register. Writing a 1 will not enable the counter. This bit is self clearing and will always read 0. 0b - No Action 1b - Clear T_EN bit for Timer Channel 3
4-31 Reserved	This read-only field is reserved and always has the value 0.

39.4.1.9 Timer Value (TVALa)

39.4.1.9.1 Address

Register	Offset
TVAL0	20h
TVAL1	30h
TVAL2	40h
TVAL3	50h

39.4.1.9.2 Function

In compare modes, these registers select the timeout period for the timer channels. In capture modes, these registers are loaded with the value of the counter when the trigger asserts.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

39.4.1.9.3 Diagram

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

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

39.4.1.9.4 Fields

Field	Function
0-31 TMR_VAL	Timer Value In compare modes, sets the timer channel start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer channel; instead the value will be loaded after the timer expires. To abort the current cycle and start a timer period with the new value, the timer channel must be disabled and enabled again. In capture modes, this register stores the inverse of the counter whenever the trigger asserts. 00000000000000000000000000000000b - Invalid load value in compare modes. 00000000000000000000000000000001-11111111111111111111111111111111b - Value to be loaded (Compare Mode) or Value of Timer (Capture Mode)

39.4.1.10 Current Timer Value (CVALa)

39.4.1.10.1 Address

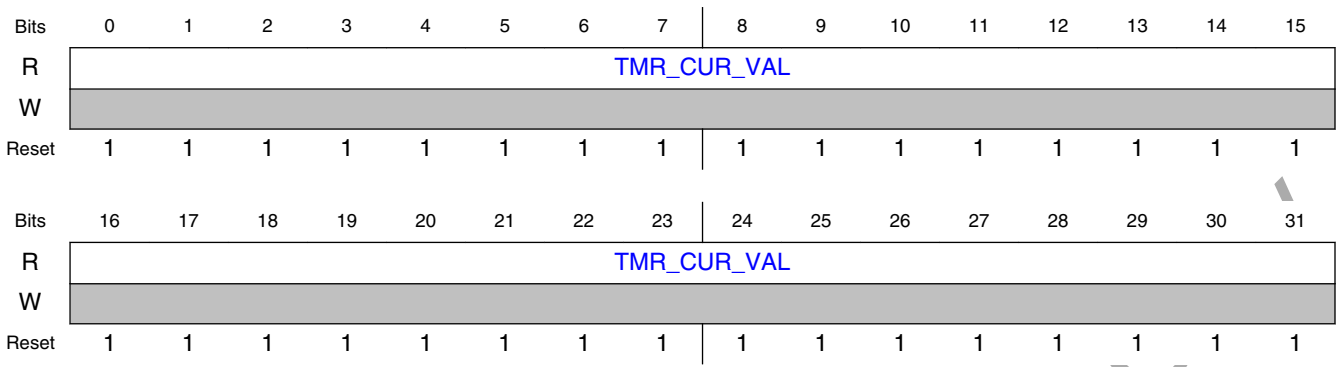
Register	Offset
CVAL0	24h
CVAL1	34h
CVAL2	44h
CVAL3	54h

39.4.1.10.2 Function

These registers indicate the current timer counter value.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

39.4.1.10.3 Diagram



39.4.1.10.4 Fields

Field	Function
0-31 TMR_CUR_VAL	Current Timer Value Represents the current timer value, if the timer is enabled.

39.4.1.11 Timer Control (TCTRLa)

39.4.1.11.1 Address

Register	Offset
TCTRL0	28h
TCTRL1	38h
TCTRL2	48h
TCTRL3	58h

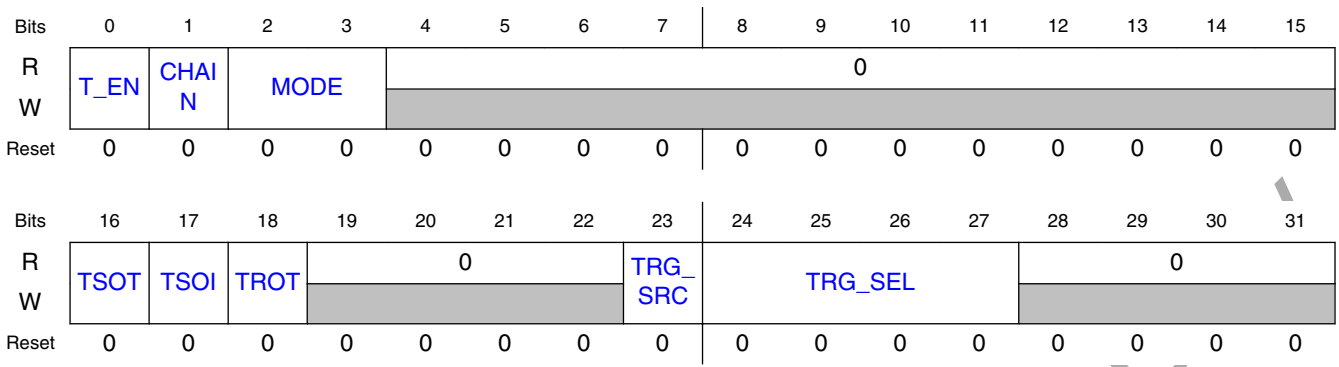
39.4.1.11.2 Function

These registers contain the control bits for each timer channel

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

39.4.1.11.3 Diagram



39.4.1.11.4 Fields

Field	Function
0 T_EN	Timer Enable Enables or disables the Timer Channel 0b - Timer Channel is disabled 1b - Timer Channel is enabled
1 CHAIN	Chain Channel When enabled, timer channel will decrement when channel N-1 trigger asserts. Channel 0 cannot be chained. 0b - Channel Chaining is disabled. Channel Timer runs independently. 1b - Channel Chaining is enabled. Timer decrements on previous channel's timeout
2-3 MODE	Timer Operation Mode Configures the Channel Timer Mode of Operation. The mode bits control how the timer decrements. See Functional Description for more details. 00b - 32-bit Periodic Counter 01b - Dual 16-bit Periodic Counter 10b - 32-bit Trigger Accumulator 11b - 32-bit Trigger Input Capture
4-15 —	This read-only field is reserved and always has the value 0.
16 TSOT	Timer Start On Trigger This bit controls when the timer starts decrementing. 0b - Timer starts to decrement immediately based on restart condition (controlled by TSOI bit) 1b - Timer starts to decrement when rising edge on selected trigger is detected
17 TSOI	Timer Stop On Interrupt This bit controls whether the channel timer will stop after it times out and when it can restart (when TSOT = 0). If TSOT = 1, then the timer will stop on timeout and will restart after a rising edge on the selected trigger is detected. If TSOT = 0, then this bit controls when the timer restarts. 0b - Timer does not stop after timeout 1b - Timer will stop after timeout and will restart after rising edge on the T_EN bit is detected (i.e. timer channel is disabled and then enabled)
18 TROT	Timer Reload On Trigger

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	When set, the LPIT timer will reload when a rising edge is detected on the selected trigger input. The trigger input is ignored if the LPIT is disabled during debug mode or DOZE mode (DOZE_EN or DBGEN = 0) 0b - Timer will not reload on selected trigger 1b - Timer will reload on selected trigger
19-22 —	This read-only field is reserved and always has the value 0
23 TRG_SRC	Trigger Source Selects between internal or external trigger sources. The final trigger is selected by TRG_SEL depending on which trigger source out of internal triggers or external triggers are selected by TRG_SRC. Refer to the chip configuration section for available external trigger options. If a channel does not have an associated external trigger then this bit for that channel should be set to 1. 0b - Trigger source selected in external 1b - Trigger source selected is the internal trigger
24-27 TRG_SEL	Trigger Select Selects the trigger to use for starting and/or reloading the LPIT timer. This field should only be changed when the LPIT timer channel is disabled. The TRG_SRC bit selects between internal and external trigger signals for each channel. The TRG_SEL bits select one trigger from the set of internal or external triggers selected by TRG_SRC. 0000-0011b - Timer channel 0 - 3 trigger source is selected. 0100-1111b - Reserved.
28-31 —	This read-only field is reserved and always has the value 0.

39.5 Functional description

39.5.1 Initialization

The following steps can be used to initialize the LPIT module

- Enable the peripheral clock by setting the M_CEN bit in the MCR register.

NOTE

- Writing to certain registers while M_CEN = 0 will lead to assertion of transfer error for that bus access. These registers are MSR, SETTEN, CLRTEN, TVAL, and TCTRL. Writing to CVAL and Reserved registers will

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

generate a transfer error irrespective of M_CEN bit value. Reads to these registers can happen irrespective of M_CEN bit value.
- There might be additional clock gating bits in the device that gate the peripheral clock to this module. Ensure software is configuring those bits in addition to M_CEN when enabling clock to this module.

- Wait for 4 peripheral clock cycles to allow time for clock synchronization and reset de-assertion.
- For each timer channel that is to be enabled, configure the timer mode of operation (MODE bits), Trigger source selection (TRG_SEL & TRG_SRC) and Trigger control bits (TROT, TSOT, TSOI bits) in the TCTRLn register.
- Configure the channels that are to be chained by setting the CHAIN bit in the corresponding channel's TCTRLn register.
- For channels configured in Compare Mode, set the timer timeout value by programming the appropriate value in TVAL register for those channels.
- Configure TIEn bits in MIER register for those channels which are required to generate interrupt on timer timeout.
- Configure the low power mode functionality of the module by setting the DBG_EN and DOZE_EN bits in the MCR register. This is common to all timer channels.
- Enable the channel timers by setting the corresponding T_EN bit in the corresponding channel's TCRTLn register.
- For channels configured in Capture Mode, the timer value can be read from TVALn register when channel timeout occurs.
- At any time, the current value of the timer for any channel can be read by reading the corresponding channel's CVALn register.
- The timer interrupt flag bits (TIFn) in MSR register get asserted on timer timeout. These bits can be cleared by writing '1' to them.

39.5.2 Timer Modes

The timer mode is configured by setting an appropriate value in the MODE bits in TCTRLn register. The timer modes supported are:

- *32-bit Periodic Counter:* In this mode the counter will load and then decrement down to zero. It will then set the timer interrupt flag and assert the output pre-trigger.
- *Dual 16-bit Periodic Counter:* In this mode, the counter will load and then the lower 16-bits will decrement down to zero, which will assert the output pre-trigger. The upper 16-bits will then decrement down to zero, which will negate the output pre-trigger and set the timer interrupt flag.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- *32-bit Trigger Accumulator:* In this mode, the counter will load on the first trigger rising edge and then decrement down to zero on each trigger rising edge. It will then set the timer interrupt flag and assert the output pre-trigger.
- *32-bit Trigger Input Capture:* In this mode, the counter will load with 0xFFFF_FFFF and then decrement down to zero. If a trigger rising edge is detected, it will store the inverse of the current counter value in the load value register, set the timer interrupt flag and assert the output pre-trigger.

The timer operation is further controlled by Trigger Control bits (TSOT, TSOI, TROT) which control the timer load, reload, start and restart of the timers.

NOTE

- The trigger output is asserted one Peripheral Timer Clock cycle later than pre-trigger output. The trigger output and the pre-trigger output de-assert at the same time.
- The pre-trigger output is asserted for two clock cycles and trigger output is asserted for one clock cycle (except in 16-bit Periodic Counter mode where both pre-trigger and trigger are asserted for many cycles depending on TMR_VAL[31:16]).

39.5.3 Trigger Control for Timers

The TSOT, TROT, TSOI and TRG_SEL, TRG_SRC bits control how the trigger input affects the timer operation. The TRG_SEL selects the input trigger for the channel from all other channel's trigger outputs. The TRG_SRC further selects between the selected internal trigger and the external trigger input to the channel.

The selected trigger affects the timer operation based on TROT, TSOI & TSOT bits. The behavior due to these bits is as follows:

- If TSOI = 1, counter stops on TIF assertion. Requires trigger (if TSOT = 1) or T_EN rising edge (if TSOT = 0), to reload and decrement. If TSOI = 0, counter does not stop after timeout.
- If TROT = 1, counter is loaded on each trigger; else, counter is loaded on every T_EN rising edge or timeout rising edge (timeout not used in Capture modes).
- If TSOT = 1, counter will start to decrement on trigger. Subsequent triggers are ignored till a counter timeout. If TSOT = 0, counter decrements immediately from the next clock edge. TSOT has no effect when channel is Chained or in Capture mode.

These bits affect the timer operation differently in different timer modes:

For Assessment Purposes Only

- In 32-bit Periodic Counter and Dual 16-bit Periodic Counter modes, all bits (TSOT, TSOI & TROT) affect the timer operation as described above.
- In 32-bit Trigger Accumulator mode, only TSOI bit controls the timer function. TROT & TSOT bits have no effect on timer operation.
- In 32-bit Input Trigger Capture mode, TSOI and TROT bits control the timer function. TSOT bit has no effect on timer operation.

39.5.4 Channel Chaining

Individual timer channels can be chained together to achieve a larger value of timeout. Chaining the timer channel causes them to work in a *'nested loop'* manner thereby leading to an effective timeout value of $TVAL_{CHn} \times (TVAL_{CHn-1} + 1)$.

The channels are chained by setting the CHAIN bit in corresponding channel's TCTRLn register. When a channel is chained, that channel's timer decrements on previous channel's timeout pulse, irrespective of the timer mode (MODE bits). The TSOT bit does not have any effect if the channel timer (Channel 'n') is chained to previous channel's timer (Channel 'n-1').

39.5.5 Detailed timing

NOTE

The timing diagrams in these sections are not *cycle-accurate* (i.e., some cycles may not be shown), but the timing diagrams do show the timer channel behavior across several clock cycles.

39.5.5.1 Mode=00: 32-bit periodic counter (compare mode)

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

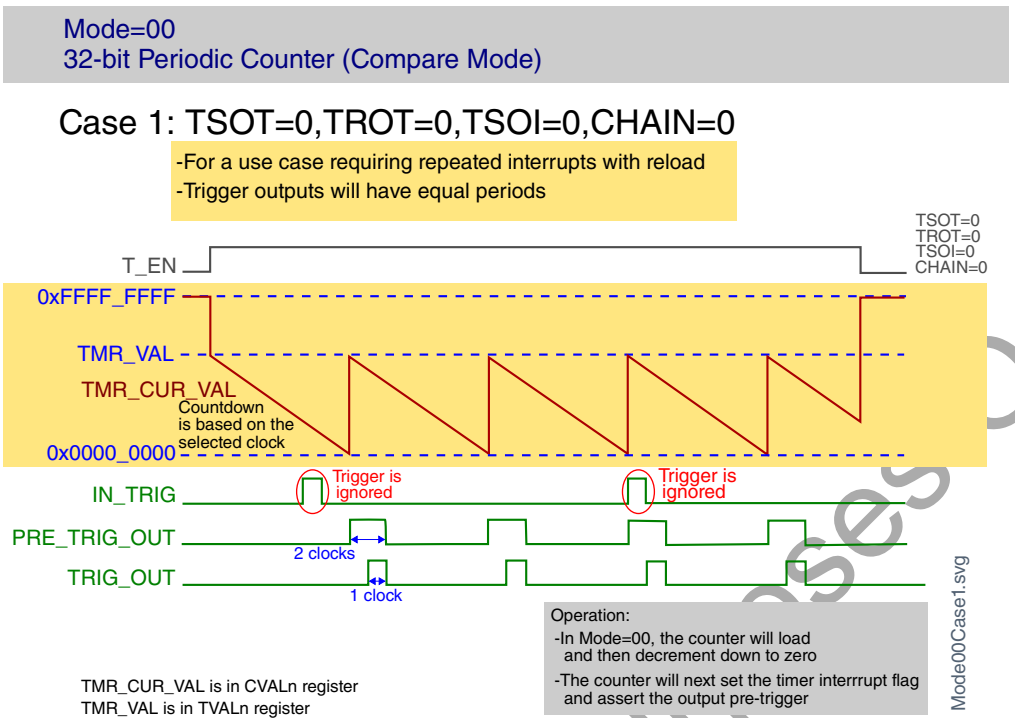


Figure 39-2. Case 1: TSOT = 0; TROT = 0; TSOI = 0; CHAIN = 0

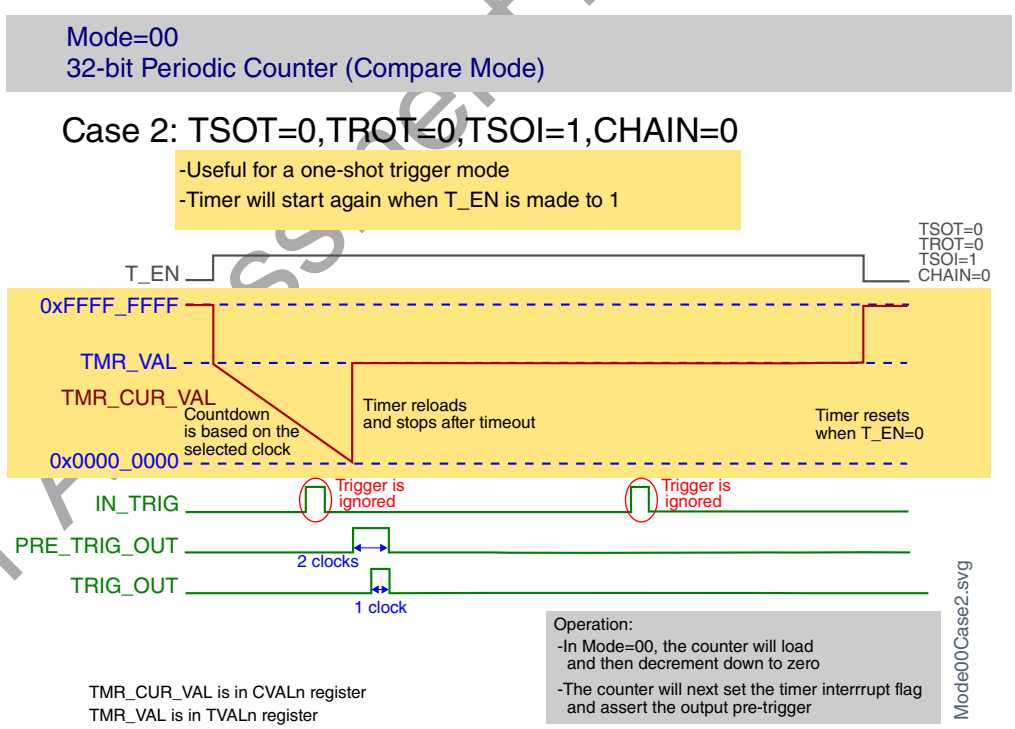


Figure 39-3. Case 2: TSOT = 0; TROT = 0; TSOI = 1; CHAIN = 0

For Preliminary Purposes Only

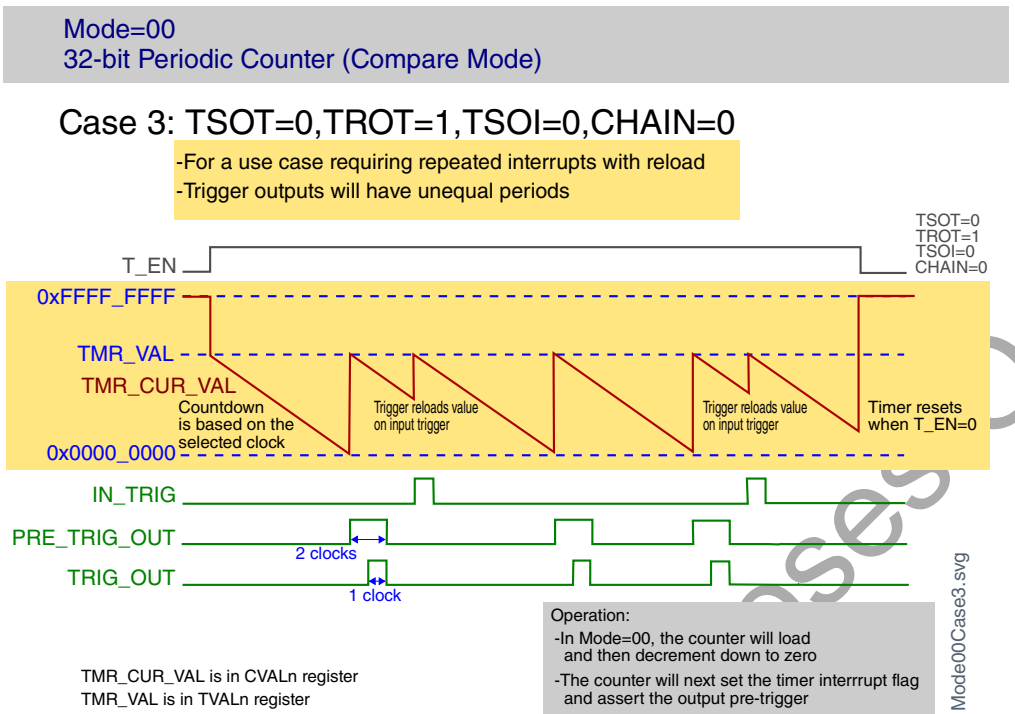


Figure 39-4. Case 3: TSOT = 0; TROT = 1; TSOI = 0; CHAIN = 0

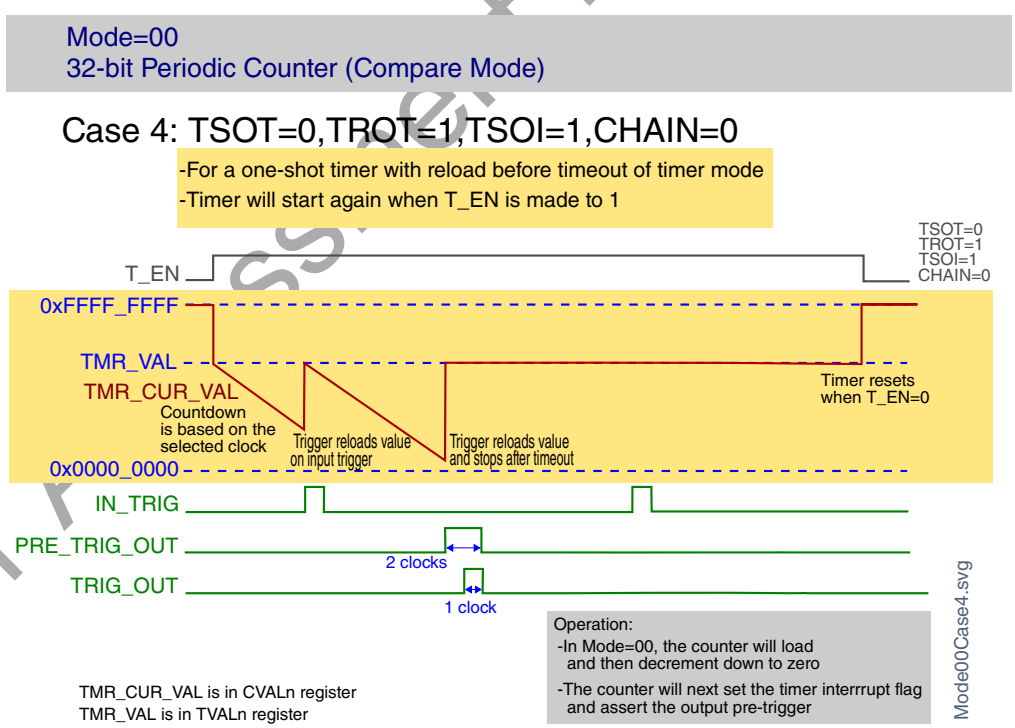


Figure 39-5. Case 4: TSOT = 0; TROT = 1; TSOI = 1; CHAIN = 0

For Preliminary Purposes Only

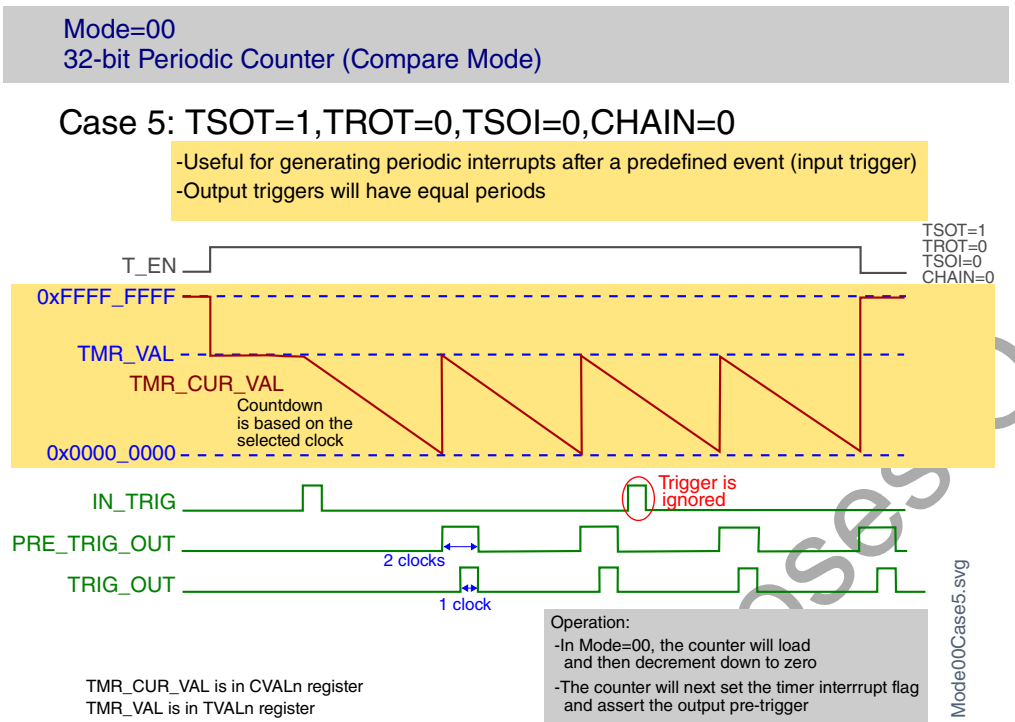


Figure 39-6. Case 5: TSOT = 1; TROT = 0; TSOI = 0; CHAIN = 0

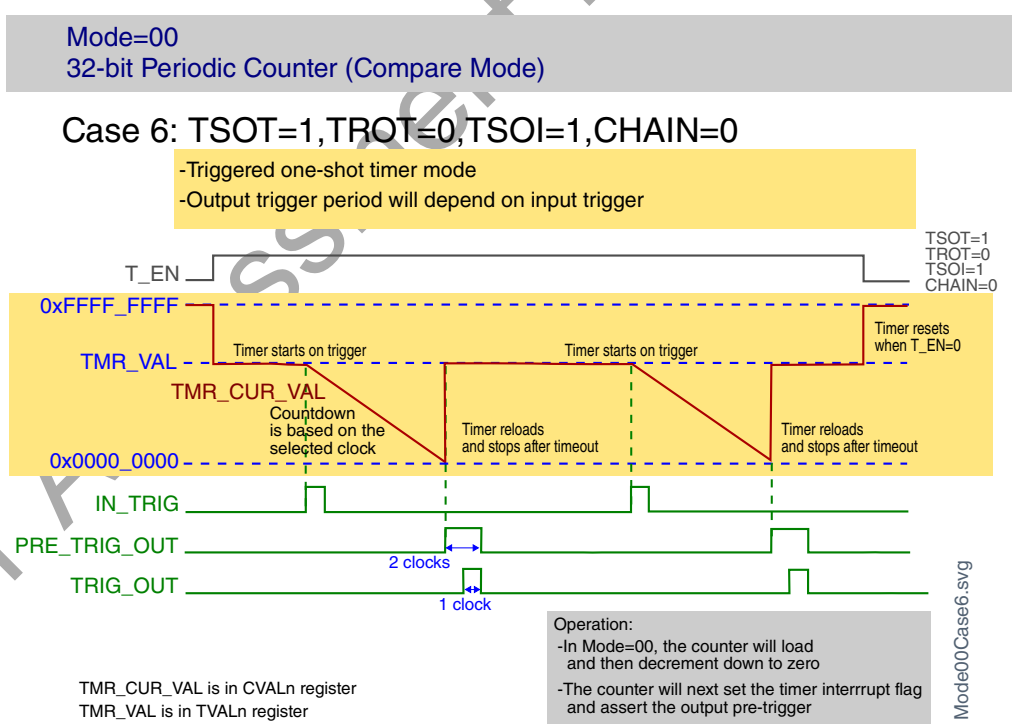


Figure 39-7. Case 6: TSOT = 1; TROT = 0; TSOI = 1; CHAIN = 0

For Access Test + Purposes Only

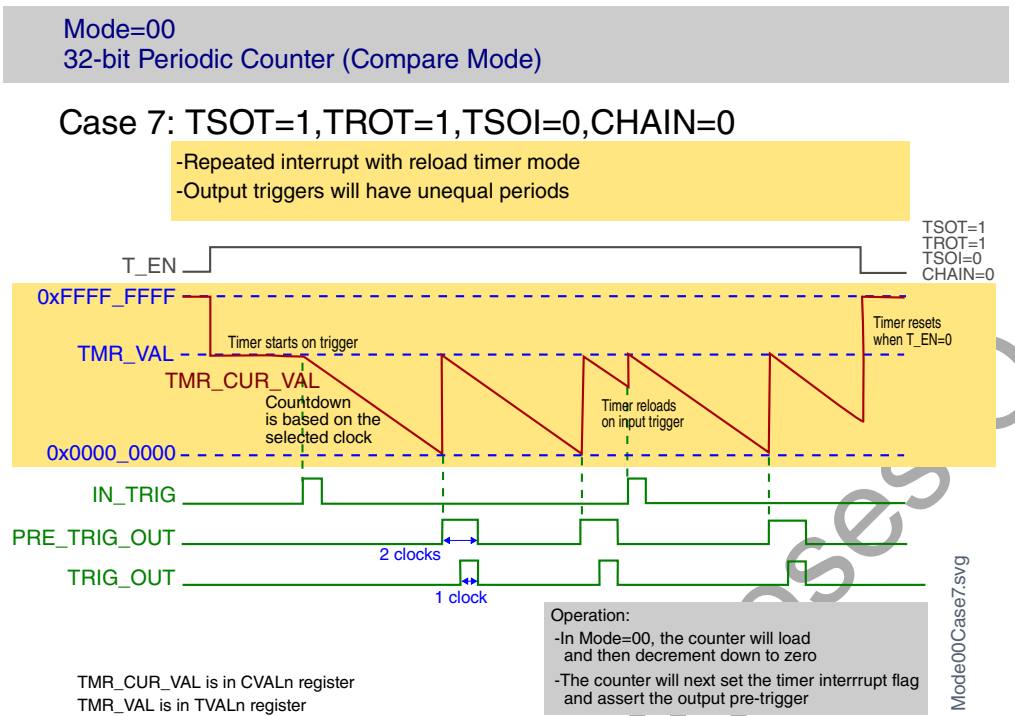


Figure 39-8. Case 7: TSOT = 1; TROT = 1; TSOI = 0; CHAIN = 0

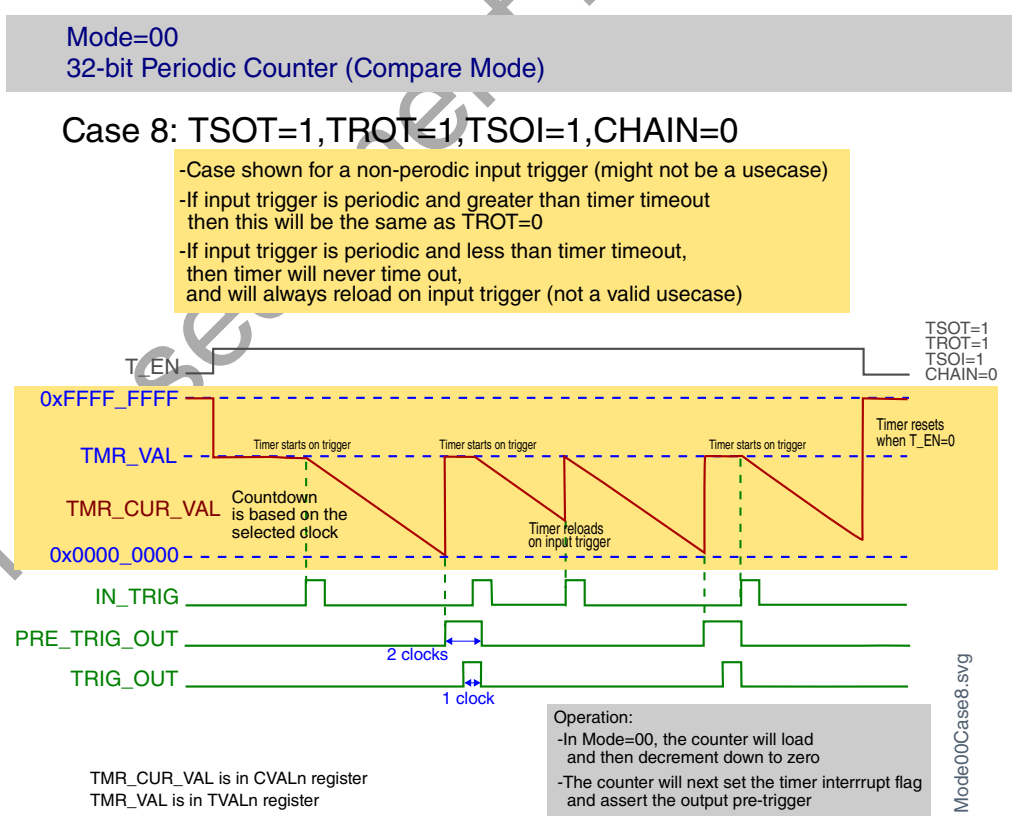


Figure 39-9. Case 8: TSOT = 1; TROT = 1; TSOI = 1; CHAIN = 0

For Reference Purposes Only

39.5.5.2 Mode=01: 16-bit dual periodic counter (compare mode)

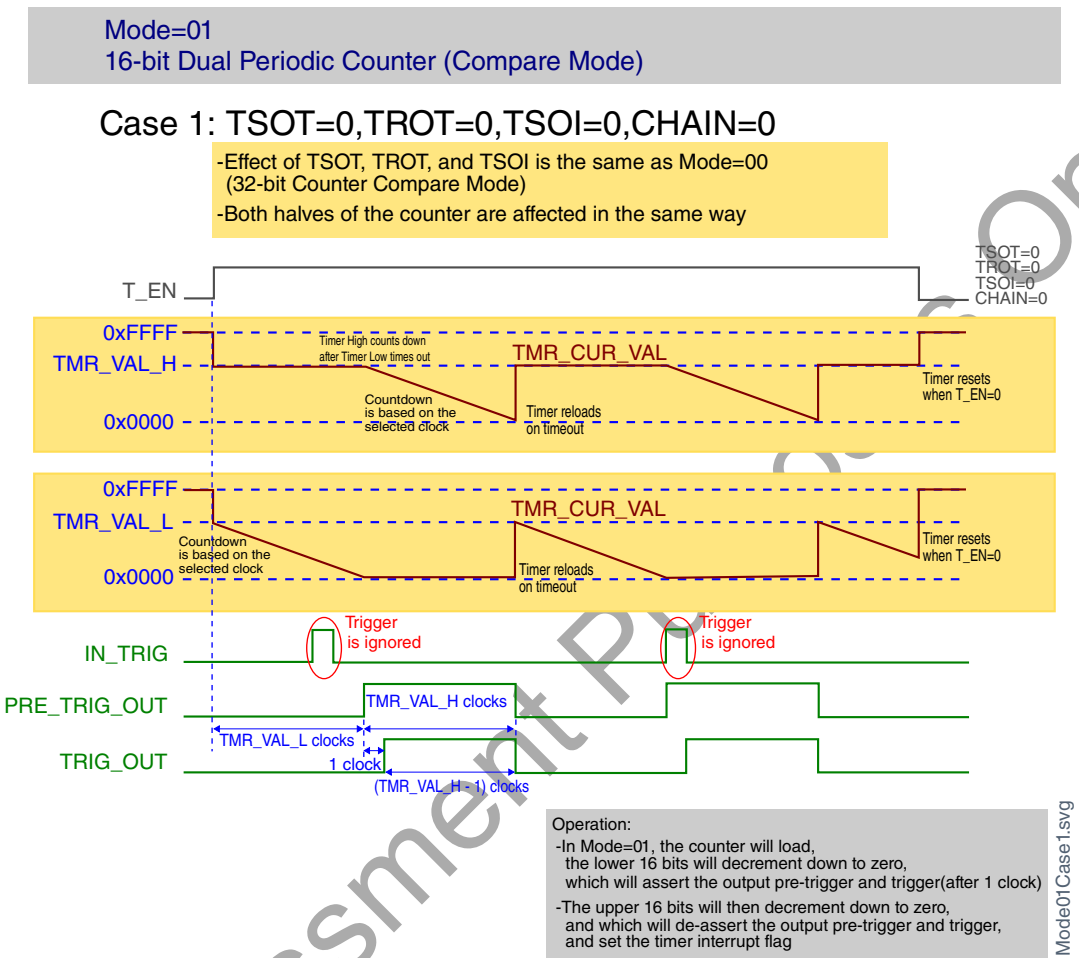


Figure 39-10. Case 1: TSOT = 0; TROT = 0; TSOI = 0; CHAIN = 0

Effect of Timer Control Bits

- Effect of timer control bits are the same as for Mode=00. Refer to the individual figures of Mode=00.
- The Timer Interrupt (timeout) asserts when {TMR_H,TMR_L} = 0x0000_0000.
- Behavior for Mode = 01 is explained in the next table.

Table 39-3. Mode=01 Timer Control Bits Function

TSOT	TROT	TSOI	Effect on Timer
0	0	0	Covered previously.
0	0	1	One shot mode. Both timers stop after first count down and timeout. Timers will not count again until T_EN is made 1 again.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Table 39-3. Mode=01 Timer Control Bits Function (continued)

TSOT	TROT	TSOI	Effect on Timer
0	1	0	Both timers will reload TMR_VAL on trigger rise edge. Output triggers will clear on reload, if asserted
0	1	1	Reloadable one shot mode: If trigger occurs before timeout, then both timers reload and count down as shown and timers stop after timeout. A trigger assertion after timeout simply reloads TMR_VAL into the timers. The timers will not count again until T_EN is set to 1 again.
1	0	0	After T_EN rises, the timers do not start until the first trigger's rising edge. Subsequent triggers will have no effect.
1	0	1	After T_EN rises, the timers do not start until the first trigger's rising edge. The timer stops counting after timeout assertion, and does not start counting again until a new trigger rising edge is detected.
1	1	0	After T_EN rises, the timers do not start until the first trigger's rising edge. Subsequent triggers will cause the timer to reload TMR_VAL into both counters. The output triggers will clear after a reload, if asserted.
1	1	1	After T_EN rises, the timers do not start until the first trigger's rising edge. The timers stops counting after timeout assertion. A trigger rising edge will cause the timers to reload and then count down.

For Assessment Purpo

Freescale Confidential Proprietary - Non-Disclosure Agreement required

39.5.5.3 Mode=10: 32-bit trigger accumulator mode

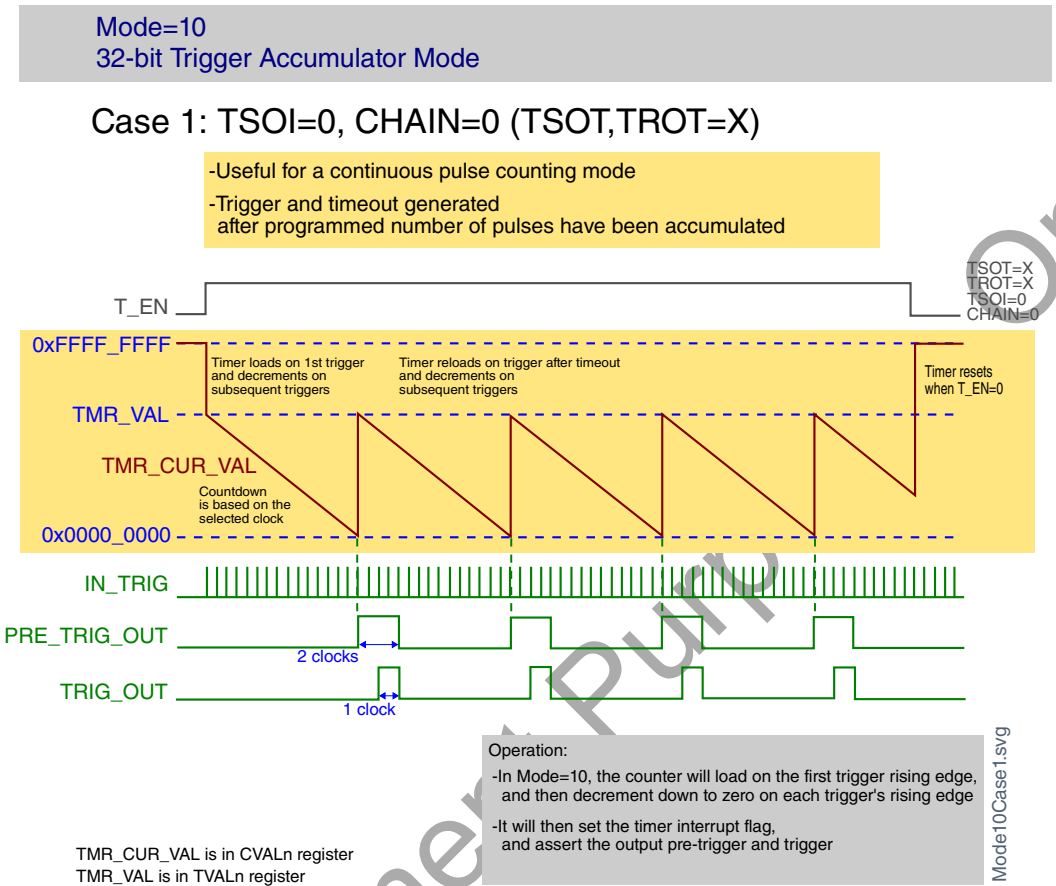


Figure 39-11. Case 1: TSOI = 0; CHAIN = 0 (TSOT, TROT = X)

For Assessment Purpose Only

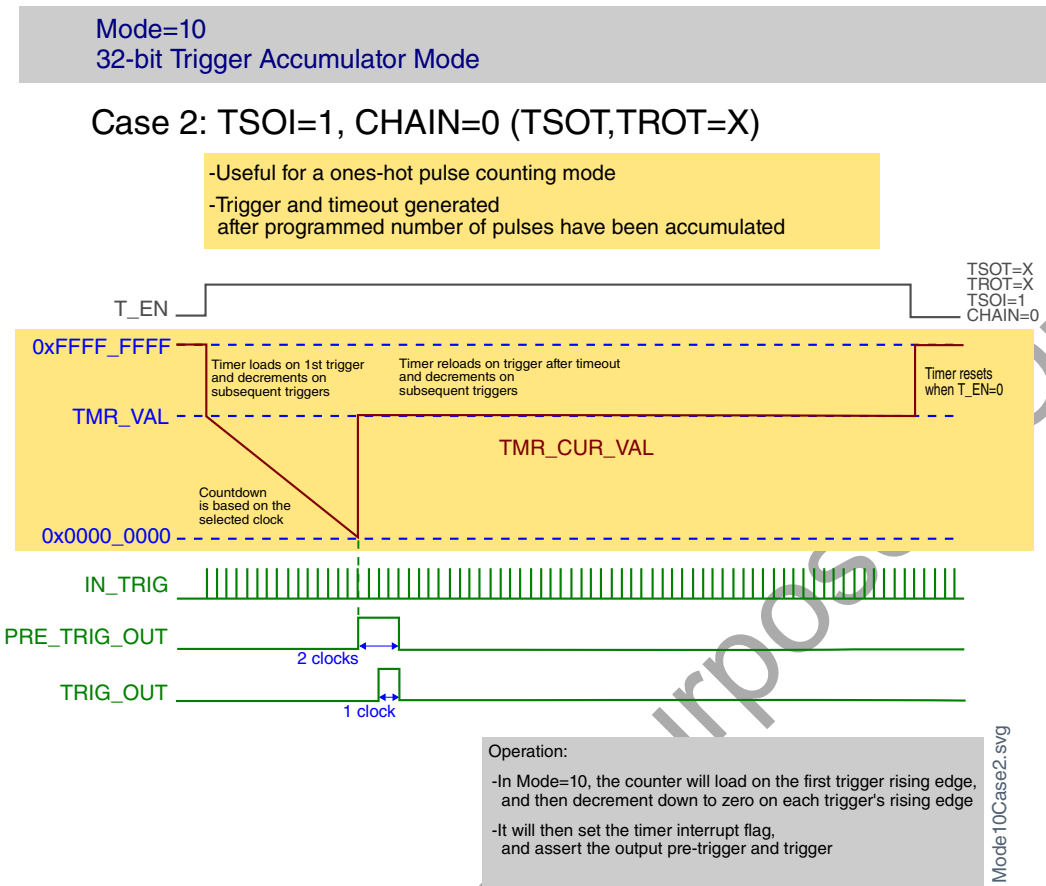


Figure 39-12. Case 2: TSOI = 1; CHAIN = 0 (TSOT, TROT = X)

For Assessment Purposes Only

39.5.5.4 Mode=11: 32-bit trigger capture mode

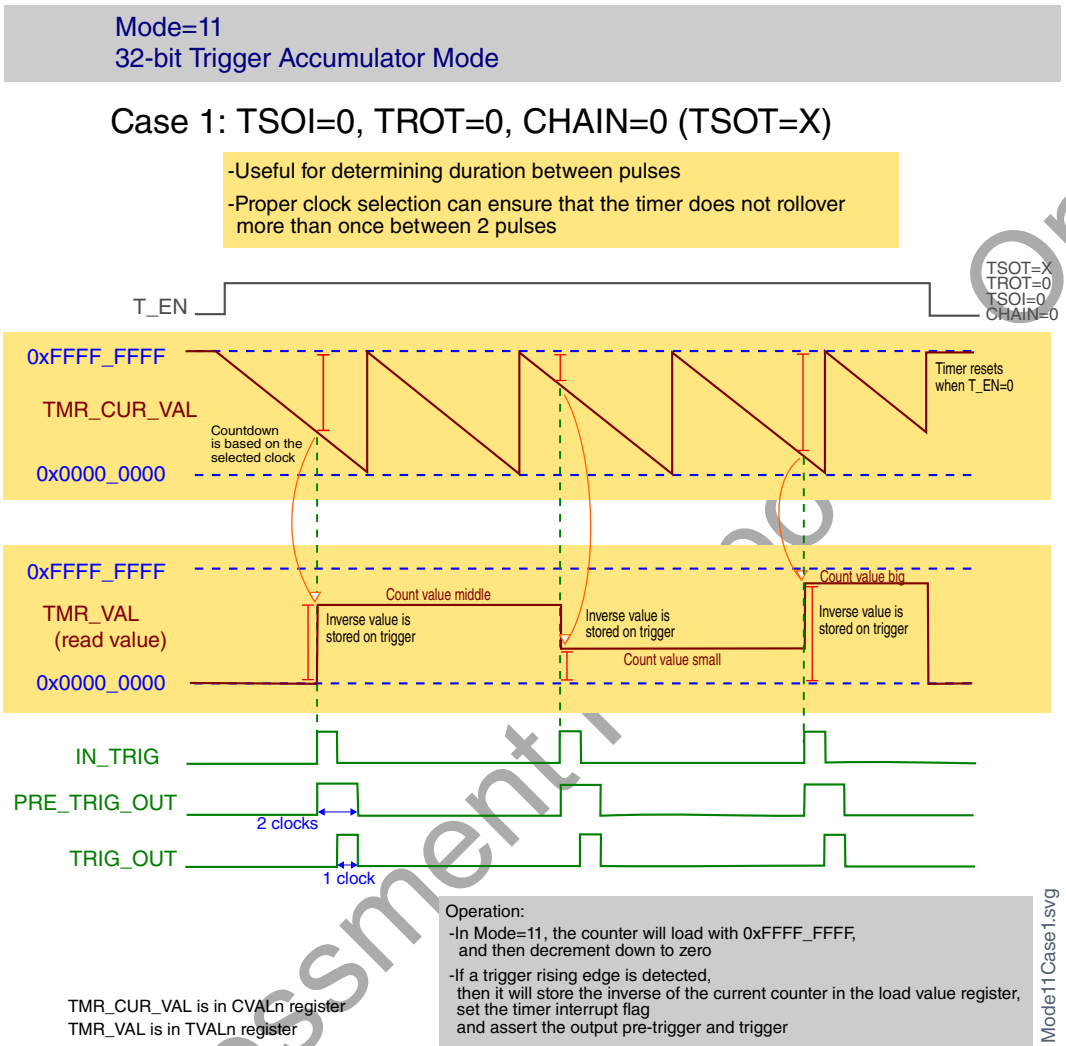


Figure 39-13. Case 1: TSOI = 0; TROT = 0; CHAIN = 0 (TSOT = X)

For Assessment Purpose Only

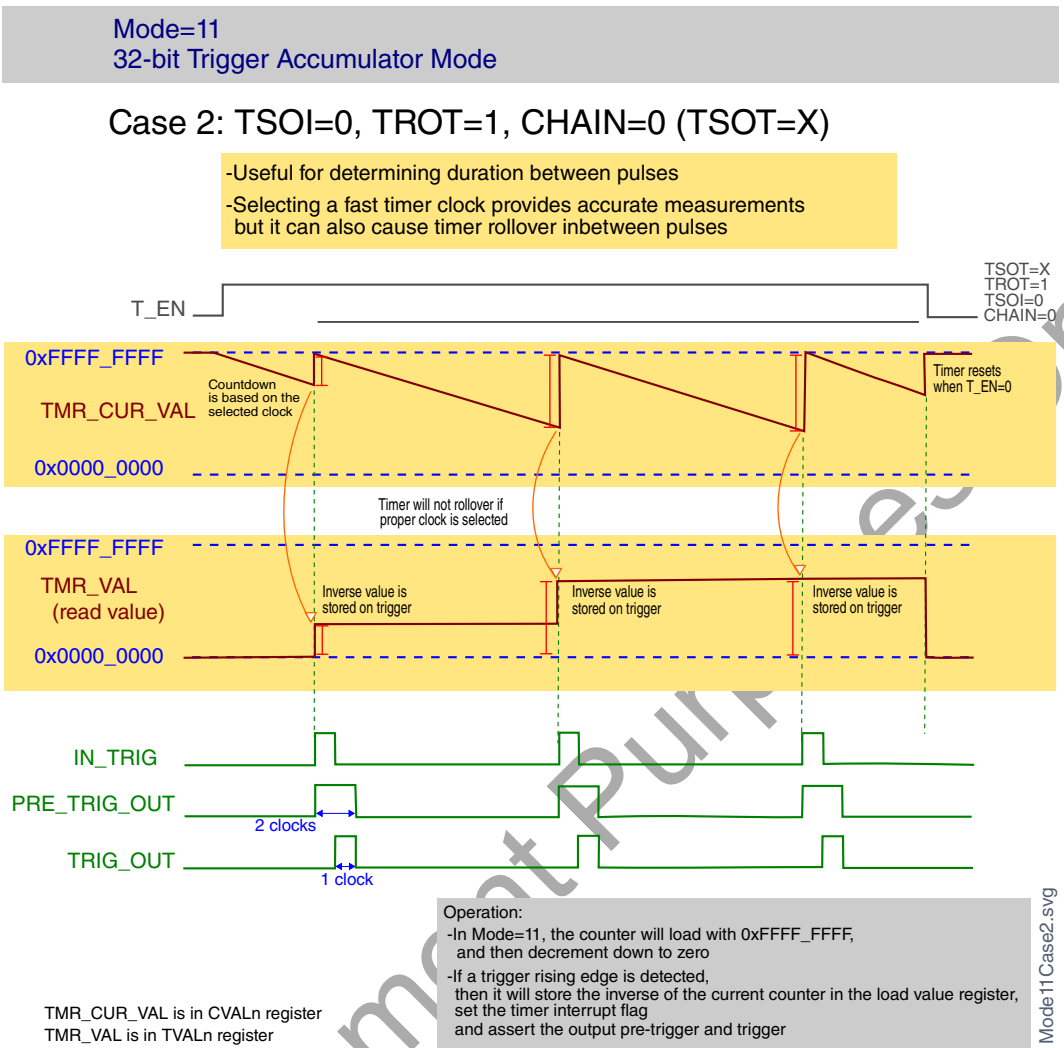


Figure 39-14. Case 2: TSOI = 0; TROT = 1; CHAIN = 0 (TSOT = X)

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

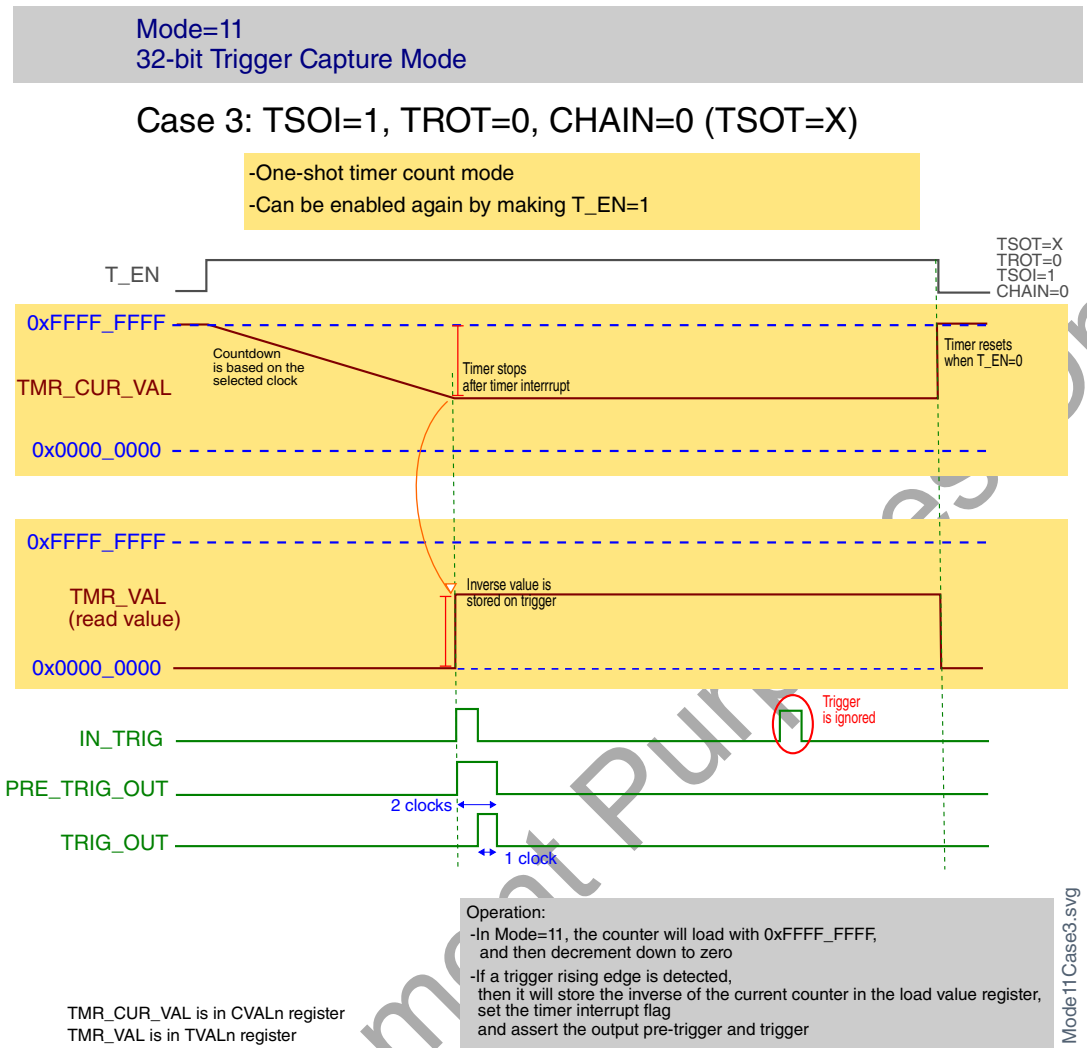


Figure 39-15. Case 3: TSOI = 1; TROT = 0; CHAIN = 0 (TSOT = X)

Case 4: TSOI = 1; TROT = 1; CHAIN = 0 (TSOT = X):

Same as previous case (Case 3), except that the timer reloads to 0xFFFF_FFFF and stops until T_EN is made 1 again. Case 4 is not a very useful case, because Case 3 covers this.

For Assessment Purposes Only

39.5.5.5 Timer chaining: effects on timing operations

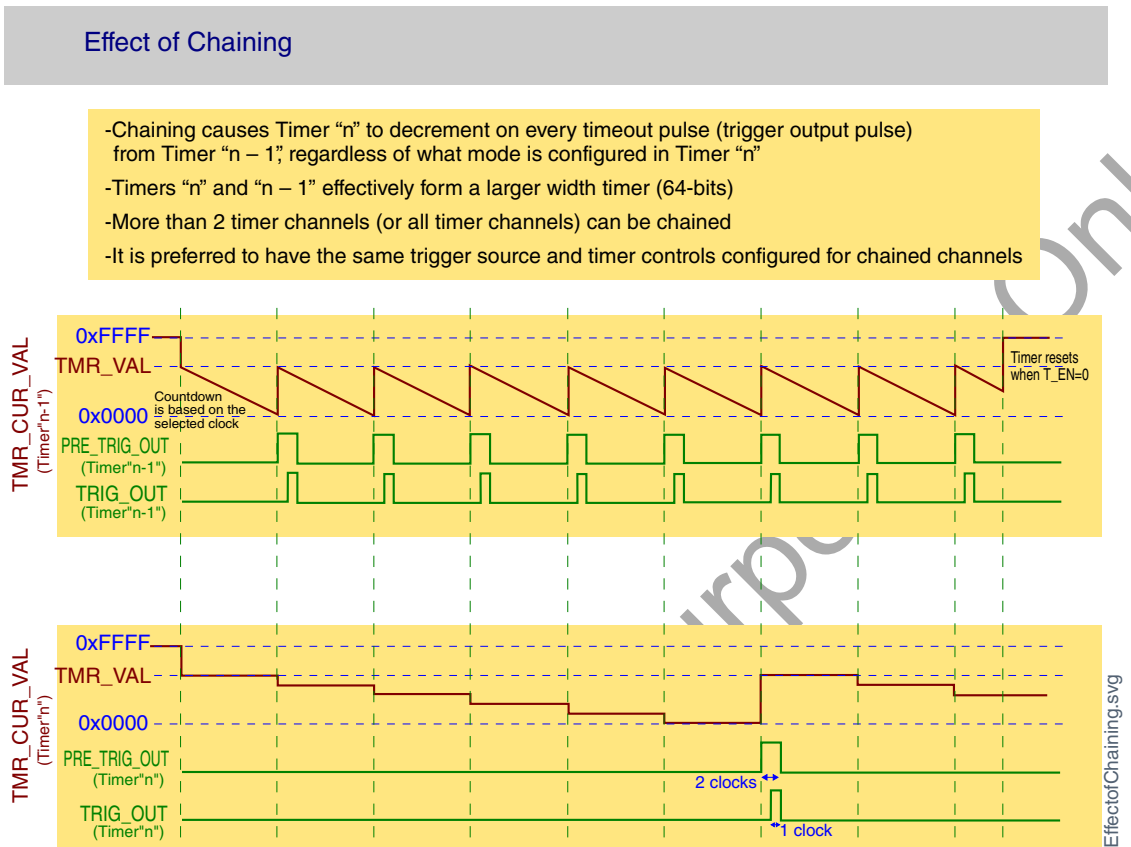


Figure 39-16. Chaining Effects

For Assessment Purpose Only

Chapter 40
Low Power Timer (LPTMR)

40.1 Chip-specific Low Power Timer (LPTMR) information

40.1.1 Instantiation Information

This device contains one LPTMR module with 1-channel, 16-bit pulse counter.

LPT/HSCMP0 pulse counting:

LPTMR_ALT0 input is the selectable source to count pulses resulting from HSCMP0 Output (LPT_ALT0 = HSCMP0 Output) via TRGMUX.

NOTE

Low leakage mode is not supported in this device.

40.1.2 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	TRGMUX output
01	1	LPTMR_ALT1 pin
10	2	LPTMR_ALT2 pin
11	3	LPTMR_ALT3 pin

For Assessment Purposes Only

40.2 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.

40.2.1 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

40.2.2 Modes of operation

The following table describes the operation of the LPTMR module in various modes.

Table 40-1. Modes of operation

Modes	Description
Run	The LPTMR operates normally.
Wait	The LPTMR continues to operate normally and can be configured to exit the low-power mode by generating an interrupt request.
Stop	The LPTMR continues to operate normally and can be configured to exit the low-power mode by generating an interrupt request.
Low-Leakage	The LPTMR continues to operate normally and can be configured to exit the low-power mode by generating an interrupt request.
Debug	The LPTMR operates normally in Pulse Counter mode, but the counter does not increment in Time Counter mode.

For Agreement Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

40.3 LPTMR signal descriptions

Table 40-2. LPTMR signal descriptions

Signal	I/O	Description
LPTMR_ALT*n*	I	Pulse Counter Input pin

40.3.1 Detailed signal descriptions

Table 40-3. 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.

40.4 Memory map and register definition

NOTE

The LPTMR registers are reset only on a POR or LVD event.
See LPTMR power and reset for more details.

40.4.1 LPTMR Register Descriptions

For Assessment Only

40.4.1.1 LPTMR Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	Low Power Timer Control Status Register (CSR)	32	RW	00000000h
4h	Low Power Timer Prescale Register (PSR)	32	RW	00000000h
8h	Low Power Timer Compare Register (CMR)	32	RW	00000000h
Ch	Low Power Timer Counter Register (CNR)	32	RW	00000000h

40.4.1.2 Low Power Timer Control Status Register (CSR)

40.4.1.2.1 Address

Register	Offset
CSR	0h

40.4.1.2.2 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TEN	TMS	TFC	TPP	TPS		TIE	TCF	TDRE	0						
W								w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

40.4.1.2.3 Fields

Field	Function
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. 0b - LPTMR is disabled and internal logic is reset. 1b - LPTMR is enabled.
1	Timer Mode Select

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
TMS	Configures the mode of the LPTMR. TMS must be altered only when the LPTMR is disabled. 0b - Time Counter mode. 1b - Pulse Counter mode.
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. 0b - CNR is reset whenever TCF is set. 1b - CNR is reset on overflow.
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. 0b - Pulse Counter input source is active-high, and the CNR will increment on the rising-edge. 1b - Pulse Counter input source is active-low, and the CNR will increment on the falling-edge.
4-5 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 configuration information about connections to these inputs. 00b - Pulse counter input 0 is selected. 01b - Pulse counter input 1 is selected. 10b - Pulse counter input 2 is selected. 11b - Pulse counter input 3 is selected.
6 TIE	Timer Interrupt Enable When TIE is set, the LPTMR Interrupt is generated whenever TCF is also set. 0b - Timer interrupt disabled. 1b - Timer interrupt enabled.
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. 0b - The value of CNR is not equal to CMR and increments. 1b - The value of CNR is equal to CMR and increments.
8 TDRE	Timer DMA Request Enable When TDRE is set, the LPTMR DMA Request is generated whenever TCF is also set and the TCF is cleared when the DMA Controller is done. 0b - Timer DMA Request disabled. 1b - Timer DMA Request enabled.
9-31 Reserved	Reserved

For R

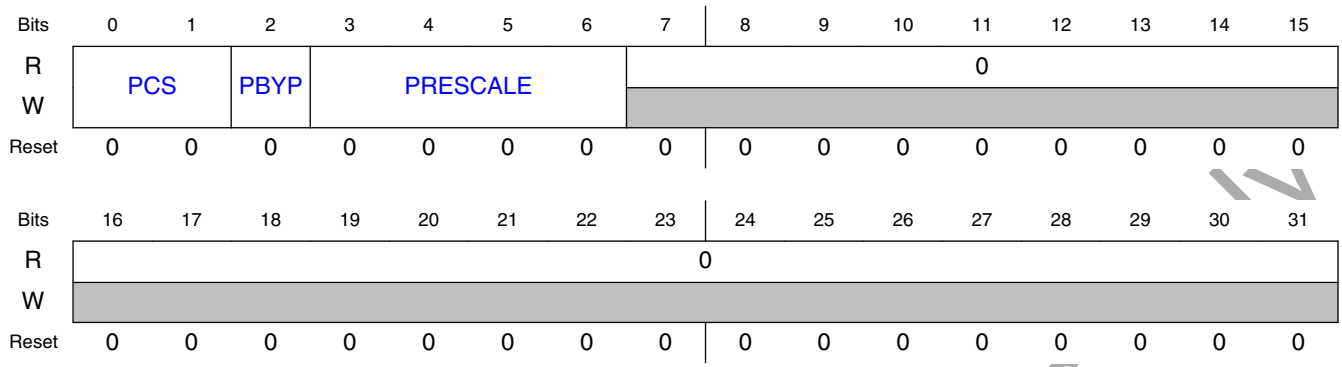
40.4.1.3 Low Power Timer Prescale Register (PSR)

40.4.1.3.1 Address

Register	Offset
PSR	4h

Freescale Confidential Proprietary - Non-Disclosure Agreement required

40.4.1.3.2 Diagram



40.4.1.3.3 Fields

Field	Function
0-1 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. 00b - Prescaler/glitch filter clock 0 selected. 01b - Prescaler/glitch filter clock 1 selected. 10b - Prescaler/glitch filter clock 2 selected. 11b - Prescaler/glitch filter clock 3 selected.
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. 0b - Prescaler/glitch filter is enabled. 1b - Prescaler/glitch filter is bypassed.
3-6 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. 0000b - Prescaler divides the prescaler clock by 2; glitch filter does not support this configuration. 0001b - Prescaler divides the prescaler clock by 4; glitch filter recognizes change on input pin after 2 rising clock edges. 0010b - Prescaler divides the prescaler clock by 8; glitch filter recognizes change on input pin after 4 rising clock edges. 0011b - Prescaler divides the prescaler clock by 16; glitch filter recognizes change on input pin after 8 rising clock edges. 0100b - Prescaler divides the prescaler clock by 32; glitch filter recognizes change on input pin after 16 rising clock edges. 0101b - Prescaler divides the prescaler clock by 64; glitch filter recognizes change on input pin after 32 rising clock edges. 0110b - Prescaler divides the prescaler clock by 128; glitch filter recognizes change on input pin after 64 rising clock edges. 0111b - Prescaler divides the prescaler clock by 256; glitch filter recognizes change on input pin after 128 rising clock edges.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	1000b - Prescaler divides the prescaler clock by 512; glitch filter recognizes change on input pin after 256 rising clock edges. 1001b - Prescaler divides the prescaler clock by 1024; glitch filter recognizes change on input pin after 512 rising clock edges. 1010b - Prescaler divides the prescaler clock by 2048; glitch filter recognizes change on input pin after 1024 rising clock edges. 1011b - Prescaler divides the prescaler clock by 4096; glitch filter recognizes change on input pin after 2048 rising clock edges. 1100b - Prescaler divides the prescaler clock by 8192; glitch filter recognizes change on input pin after 4096 rising clock edges. 1101b - Prescaler divides the prescaler clock by 16,384; glitch filter recognizes change on input pin after 8192 rising clock edges. 1110b - Prescaler divides the prescaler clock by 32,768; glitch filter recognizes change on input pin after 16,384 rising clock edges. 1111b - Prescaler divides the prescaler clock by 65,536; glitch filter recognizes change on input pin after 32,768 rising clock edges.
7-31 Reserved	Reserved

40.4.1.4 Low Power Timer Compare Register (CMR)

40.4.1.4.1 Address

Register	Offset
CMR	8h

40.4.1.4.2 Diagram

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

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

40.4.1.4.3 Fields

Field	Function
0-15 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.
16-31 —	Reserved

40.4.1.5 Low Power Timer Counter Register (CNR)

40.4.1.5.1 Address

Register	Offset
CNR	Ch

40.4.1.5.2 Diagram

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

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

40.4.1.5.3 Fields

Field	Function
0-15 COUNTER	Counter ValueThe CNR returns the current value of the LPTMR counter at the time this register was last written.
16-31 —	Reserved

Freescale Confidential Proprietary - Non-Disclosure Agreement required

40.5 Functional description

40.5.1 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.

40.5.2 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 may 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.

40.5.3 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.

For Assessment Purposes Only

NOTE

The prescaler/glitch filter configuration must not be altered when the LPTMR is enabled.

40.5.3.1 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.

40.5.3.2 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.

40.5.3.3 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

40.5.3.4 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.

40.5.4 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.

40.5.5 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.

When the core is halted in Debug mode:

- If configured for Pulse Counter mode, the CNR continues incrementing.
- If configured for Time Counter mode, the CNR stops incrementing.

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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

40.5.6 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.

40.5.7 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 41
Real Time Clock (RTC)

41.1 Chip-specific Real Time Clock (RTC)information

41.1.1 RTC Instantiation

NOTE

The wakeup pin is not available for RTC on this device, therefore the related register bitfields are not applicable (e.g. RTC_CR[WPS], RTC_CR[WPE], and RTC_IER[WPON]).

NOTE

Also there is no integrated capacitor for this device, therefore no tunable capacitors (included in the crystal oscillator) can be configured by software.

41.2 Introduction

41.2.1 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
- Option to increment prescaler using the LPO (prescaler increments by 32 every clock edge)
- Register write protection

For Assessment Purposes Only

- Lock register requires POR or software reset to enable write access
- Configurable 1, 2, 4, 8, 16, 32, 64 or 128 Hz square wave output with optional interrupt

41.2.2 Modes of operation

The RTC remains functional in all low power modes and can generate an interrupt to exit any low power mode.

41.2.3 RTC signal descriptions

Table 41-1. RTC signal descriptions

Signal	Description	I/O
RTC_CLKOUT	Prescaler square-wave output or 32kHz crystal clock	O

41.2.3.1 RTC clock output

The RTC_CLKOUT signal can output either a square wave prescaler output (configurable to 1, 2, 4, 8, 16, 32, 64 or 128 Hz) or the 32 kHz crystal clock.

41.3 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RTC memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	RTC Time Seconds Register (RTC_TSR)	32	R/W	0000_0000h	41.3.1/987
4	RTC Time Prescaler Register (RTC_TPR)	32	R/W	0000_0000h	41.3.2/987
8	RTC Time Alarm Register (RTC_TAR)	32	R/W	0000_0000h	41.3.3/988
C	RTC Time Compensation Register (RTC_TCR)	32	R/W	0000_0000h	41.3.4/988
10	RTC Control Register (RTC_CR)	32	R/W	0000_0000h	41.3.5/990
14	RTC Status Register (RTC_SR)	32	R/W	0000_0001h	41.3.6/992
18	RTC Lock Register (RTC_LR)	32	R/W	0000_00FFh	41.3.7/993
1C	RTC Interrupt Enable Register (RTC_IER)	32	R/W	0000_0007h	41.3.8/994

41.3.1 RTC Time Seconds Register (RTC_TSR)

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

Bit	31	30	29	28	27	26	25	24	23	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).

41.3.2 RTC Time Prescaler Register (RTC_TPR)

Address: 0h base + 4h offset = 4h

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

RTC_TPR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RTC_TPR field descriptions (continued)

Field	Description
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.

41.3.3 RTC Time Alarm Register (RTC_TAR)

Address: 0h base + 8h offset = 8h

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

RTC_TAR field descriptions

Field	Description
TAR	Time Alarm Register When the time counter is enabled, the SR[TAF] is set whenever the TAR[TAR] equals the TSR[TSR] and the TSR[TSR] increments. Writing to the TAR clears the SR[TAF].

41.3.4 RTC Time Compensation Register (RTC_TCR)

Address: 0h base + Ch offset = Ch

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

RTC_TCR field descriptions

Field	Description
31–24 CIC	Compensation Interval Counter Current value of the compensation interval counter. If the compensation interval counter equals zero then it is loaded with the contents of the CIR. If the CIC does not equal zero then it is decremented once a second.
23–16 TCV	Time Compensation Value Current value used by the compensation logic for the present second interval. Updated once a second if the CIC equals 0 with the contents of the TCR field. If the CIC does not equal zero then it is loaded with zero (compensation is not enabled for that second increment).

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RTC_TCR field descriptions (continued)

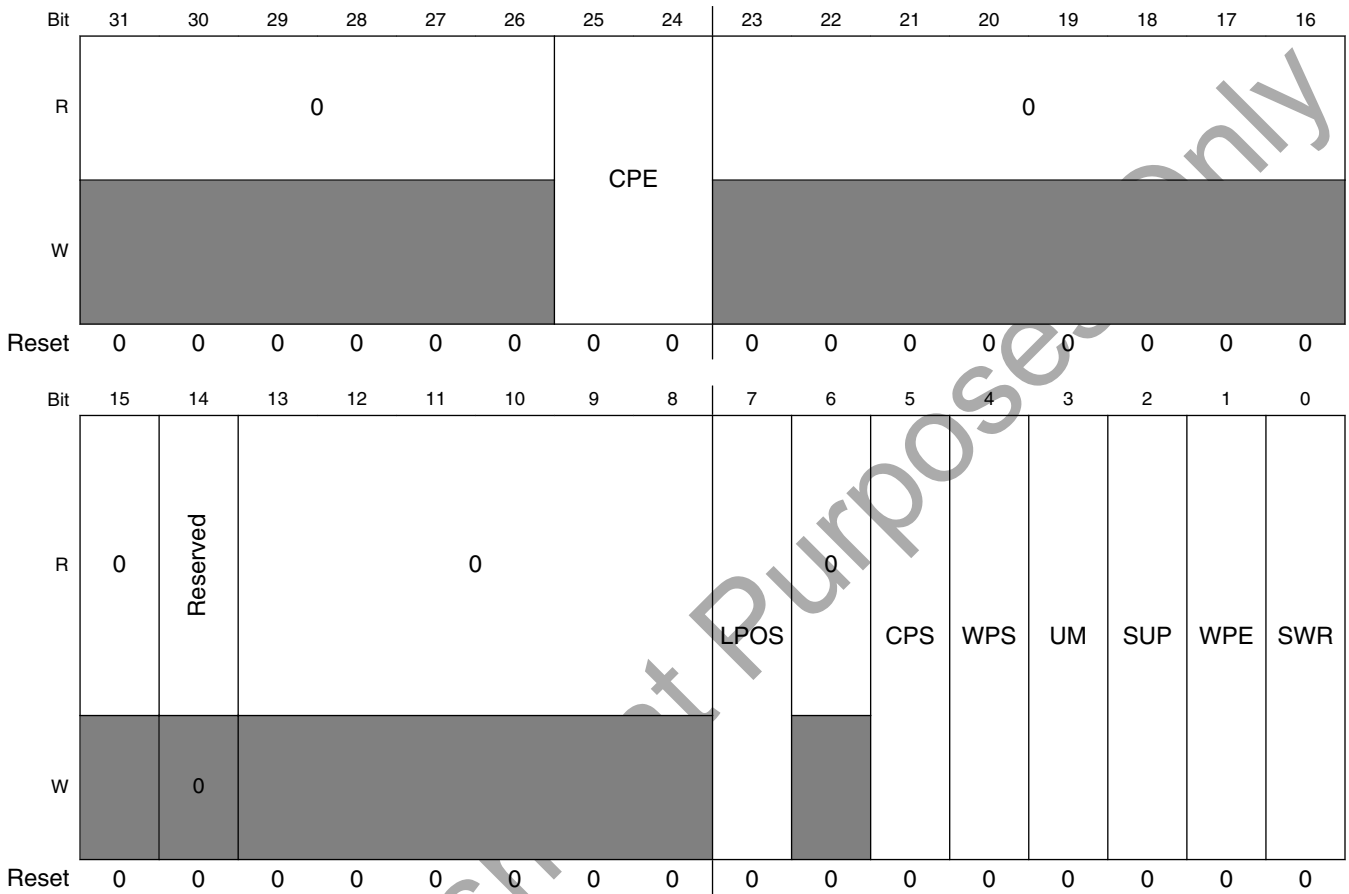
Field	Description
15–8 CIR	Compensation Interval Register Configures the compensation interval in seconds from 1 to 256 to control how frequently the TCR should adjust the number of 32.768 kHz cycles in each second. The value written should be one less than the number of seconds. For example, write zero 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.

For Assessment Purp

Freescale Confidential Proprietary - Non-Disclosure Agreement required

41.3.5 RTC Control Register (RTC_CR)

Address: 0h base + 10h offset = 10h



Student Purposes Only

RTC_CR field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 CPE	Clock Pin Enable **NOTE:** The CPE field should be configured to 01 or 11 (i.e. CPE[0] = 1), if we want the RTC_CLKOUT signal as output. 00 RTC_CLKOUT is disabled. 01 RTC_CLKOUT is enabled on pin PTE0. 10 RTC_CLKOUT is enabled on pin PTE26. 11 Reserved.
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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

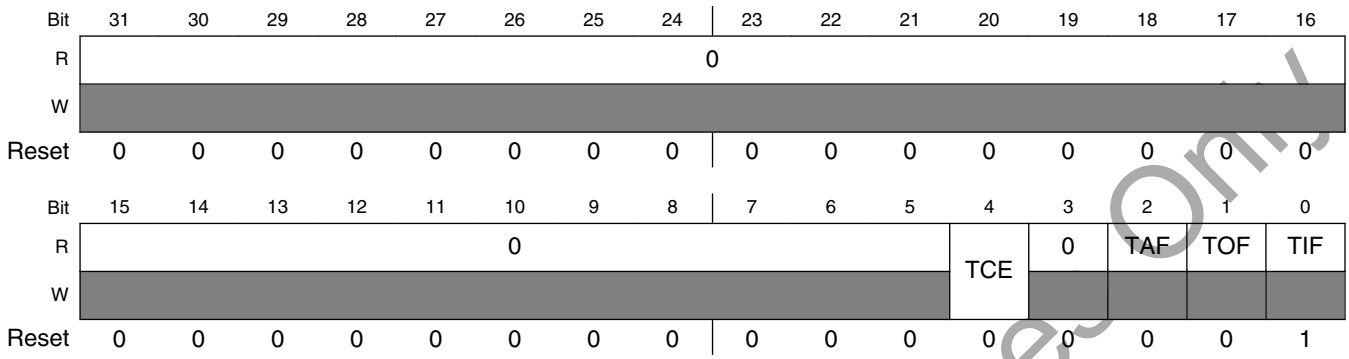
RTC_CR field descriptions (continued)

Field	Description
13–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 LPOS	LPO Select When set, the RTC prescaler increments using the LPO 1kHz clock and not the RTC 32kHz crystal clock. The LPO increments the prescaler from bit TPR[5] (TPR[4:0] are ignored), supporting close to 1 second increment of the seconds register. Although compensation is supported when clocked from the LPO, TCR[4:0] of the compensation register are also ignored and only TCR[7:5] set the compensation value (can overflow after 1020 to 1027 cycles). 0 RTC prescaler increments using 32kHz crystal. 1 RTC prescaler increments using 1kHz LPO, bits [4:0] of the prescaler are bypassed.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 CPS	Clock Pin Select 0 The prescaler output clock (as configured by TSIC) is output on RTC_CLKOUT. 1 The RTC 32kHz crystal clock is output on RTC_CLKOUT.
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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

41.3.6 RTC Status Register (RTC_SR)

Address: 0h base + 14h offset = 14h



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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

41.3.7 RTC Lock Register (RTC_LR)

Address: 0h base + 18h offset = 18h

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

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

RTC_LR field descriptions

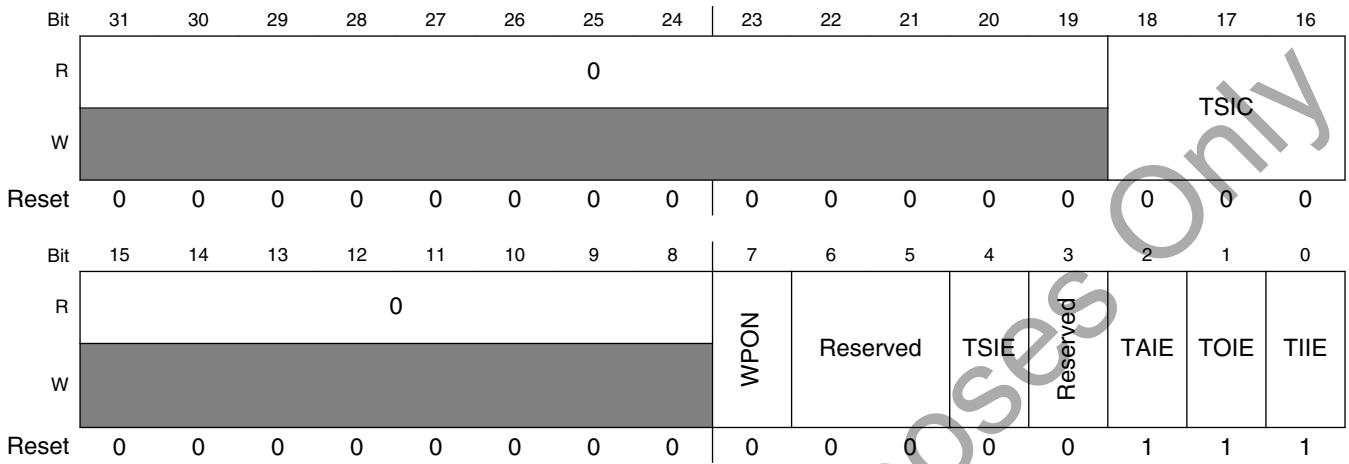
Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
6 LRL	Lock Register Lock After being cleared, this bit can be set only by POR or software reset. 0 Lock Register is locked and writes are ignored. 1 Lock Register is not locked and writes complete as normal.
5 SRL	Status Register Lock After being cleared, this bit can be set only by POR or software reset. 0 Status Register is locked and writes are ignored. 1 Status Register is not locked and writes complete as normal.
4 CRL	Control Register Lock After being cleared, this bit can only be set by POR. 0 Control Register is locked and writes are ignored. 1 Control Register is not locked and writes complete as normal.
3 TCL	Time Compensation Lock After being cleared, this bit can be set only by POR or software reset. 0 Time Compensation Register is locked and writes are ignored. 1 Time Compensation Register is not locked and writes complete as normal.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

Freescale Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

41.3.8 RTC Interrupt Enable Register (RTC_IER)

Address: 0h base + 1Ch offset = 1Ch



Purposes Only

RTC_IER field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 TSIC	Timer Seconds Interrupt Configuration Configures the frequency of the RTC Seconds interrupt and the RTC_CLKOUT prescaler output. This field should only be altered when TSIE is clear. 000 1 Hz. 001 2 Hz. 010 4 Hz. 011 8 Hz. 100 16 Hz. 101 32 Hz. 110 64 Hz. 111 128 Hz.
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

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RTC_IER field descriptions (continued)

Field	Description
	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 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.

For Assessment Pur

41.4 Functional description

41.4.1 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. Alternatively, the time counter can be clocked by the LPO and the prescaler will increment by 32 for each LPO clock.

The power-on-reset signal initializes all RTC registers to their default state. A software reset bit can also initialize all RTC registers.

41.4.1.1 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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

41.4.1.2 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.

41.4.2 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. There is also the option to clock the prescaler using a 1 kHz LPO that increments the prescaler by 32 on every 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 (or 1 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

41.4.3 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.

When the prescaler is configured to increment using the 1 kHz LPO, the effective compensation value is divided by 32 and can only adjust the number of clock cycles between -4 and +3.

41.4.4 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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].

41.4.5 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].

41.4.6 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.

41.4.7 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. The frequency of the seconds interrupt defaults to 1 Hz, but can instead be configured to trigger every 2, 4, 8, 16, 32, 64 or 128 Hz. This interrupt is optional and may not be implemented on all devices.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Chapter 42
Low Power Serial Peripheral Interface (LPSPI)

42.1 Chip-specific Low Power Serial Peripheral Interface (LPSPI) information

42.1.1 Instantiation Information

This device contains three LPSPI modules.

Table 42-1. LPSPI Configuration

LPSPI Feature	LPSPI0	LPSPI1	LPSPI2
TX FIFO (word)	4	4	4
RX FIFO (word)	4	4	4
Chip Selects	4	4	4

NOTE

The exact number of chip select for each module is depending on the package, not all of the chip selects are available on different packages.

NOTE

LPSPI2 does not support any TRGMUX related feature, say HREQ source from TRGMUX or any trigger to TRGMUX.

NOTE

Low leakage mode is not supported in this device.

42.2 Introduction

For Assessment Purposes Only

42.2.1 Overview

The LPSPI is a low power Serial Peripheral Interface (SPI) module that supports an efficient interface to an SPI bus as a master and/or a slave. The LPSPI can continue operating in stop modes provided an appropriate clock is available and is designed for low CPU overhead with DMA offloading of FIFO register accesses.

42.2.2 Features

The LPSPI supports the following features:

- Word size = 32 bits
- Command/transmit FIFO of 4 words.
- Receive FIFO of 4 words.
- Host request input can be used to control the start time of an SPI bus transfer.

For Assessment Purposes Only

42.2.3 Block Diagram

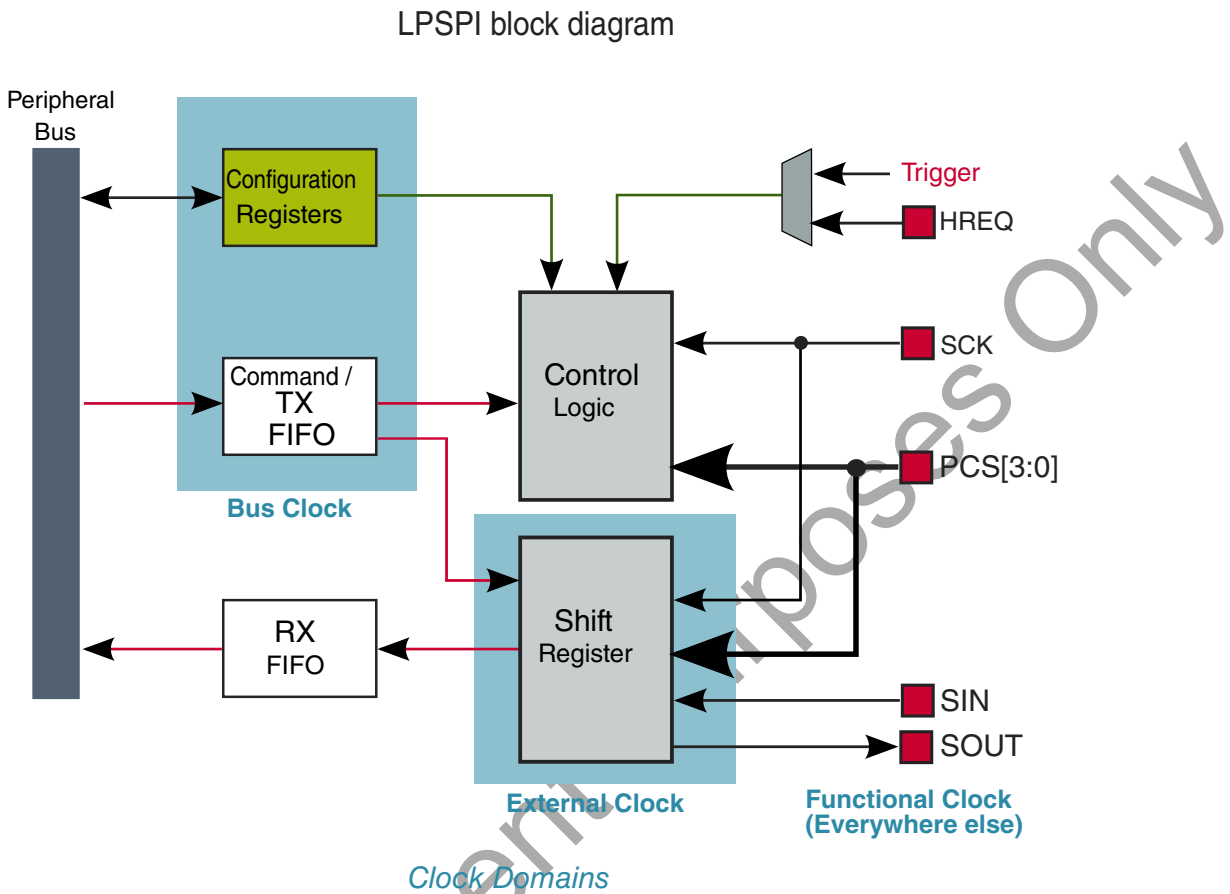


Figure 42-1. Block Diagram

42.2.4 Modes of operation

The LPSPI module supports the chip modes described in the following table.

Table 42-2. Chip modes supported by the LPSPI module

Chip mode	LPSPI Operation
Run	Normal operation
Stop/Wait	Can continue operating if the Doze Enable bit (MCR[DOZEN]) is clear and the LPSPI is using an external or internal clock source, which remains operating during stop/wait modes.
Low Leakage Stop (LLS/VLLS modes)	The Doze Enable (MCR[DOZEN]) bit is ignored and the LPSPI will wait for the current transfer to finish any pending operation, before the LPSPI acknowledges entry into low leakage mode.
Debug (the core is in Debug/Halted mode)	Can continue operating if the Debug Enable bit (MCR[DBGE]) is set.

Assessment Purposes Only

42.2.5 Signal Descriptions

Signal	Description	I/O
SCK	Serial clock. Input in slave mode, output in master mode.	I/O
PCS[0]	Peripheral Chip Select. Input in slave mode, output in master mode.	I/O
PCS[1] / HREQ	Peripheral Chip Select or Host Request. Host Request pin is selected when HREN=1 and HRSEL=0. Input in either slave mode or when used as Host Request, output in master mode.	I/O
PCS[2] / DATA[2]	Peripheral Chip Select or data pin 2 during quad-data transfers. Input in slave mode, output in master mode, input in quad-data receive transfers, output in quad-data transmit transfers.	I/O
PCS[3] / DATA[3]	Peripheral Chip Select or data pin 3 during quad-data transfers. Input in slave mode, output in master mode, input in quad-data receive transfers, output in quad-data transmit transfers.	I/O
SOUT / DATA[0]	Serial Data Output. Can be configured as serial data input signal. Used as data pin 0 in quad-data and dual-data transfers.	I/O
SIN / DATA[1]	Serial Data Input. Can be configured as serial data output signal. Used as data pin 1 in quad-data and dual-data transfers.	I/O

For Assessi

42.3 Memory Map and Registers

42.3.1 LPSPI Register Descriptions

42.3.1.1 LPSPI Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	RO	01000004h

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Offset	Register	Width (In bits)	Access	Reset value
4h	Parameter (PARAM)	32	RO	00000202h
10h	Control (CR)	32	RW	00000000h
14h	Status (SR)	32	W1C	00000001h
18h	Interrupt Enable (IER)	32	RW	00000000h
1Ch	DMA Enable (DER)	32	RW	00000000h
20h	Configuration 0 (CFGR0)	32	RW	00000000h
24h	Configuration 1 (CFGR1)	32	RW	00000000h
30h	Data Match 0 (DMR0)	32	RW	00000000h
34h	Data Match 1 (DMR1)	32	RW	00000000h
40h	Clock Configuration (CCR)	32	RW	00000000h
58h	FIFO Control (FCR)	32	RW	00000000h
5Ch	FIFO Status (FSR)	32	RO	00000000h
60h	Transmit Command (TCR)	32	RW	0000001Fh
64h	Transmit Data (TDR)	32	WO	00000000h
70h	Receive Status (RSR)	32	RO	00000002h
74h	Receive Data (RDR)	32	RO	00000000h

42.3.1.2 Version ID (VERID)

42.3.1.2.1 Address

Register	Offset
VERID	0h

42.3.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	0	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	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Assessment Pu

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.2.3 Fields

Field	Function
31-24 MAJOR	Major Version Number This read-only field returns the major version number for the module specification.
23-16 MINOR	Minor Version Number This read-only field returns the minor version number for the module specification.
15-0 FEATURE	Module Identification Number This read-only field returns the feature set number. 0000000000000100b - Standard feature set supporting 32-bit shift register.

42.3.1.3 Parameter (PARAM)

42.3.1.3.1 Address

Register	Offset
PARAM	4h

42.3.1.3.2 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	RXFIFO								TXFIFO							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

42.3.1.3.3 Fields

Field	Function
31-16 —	Reserved
15-8	Receive FIFO Size

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

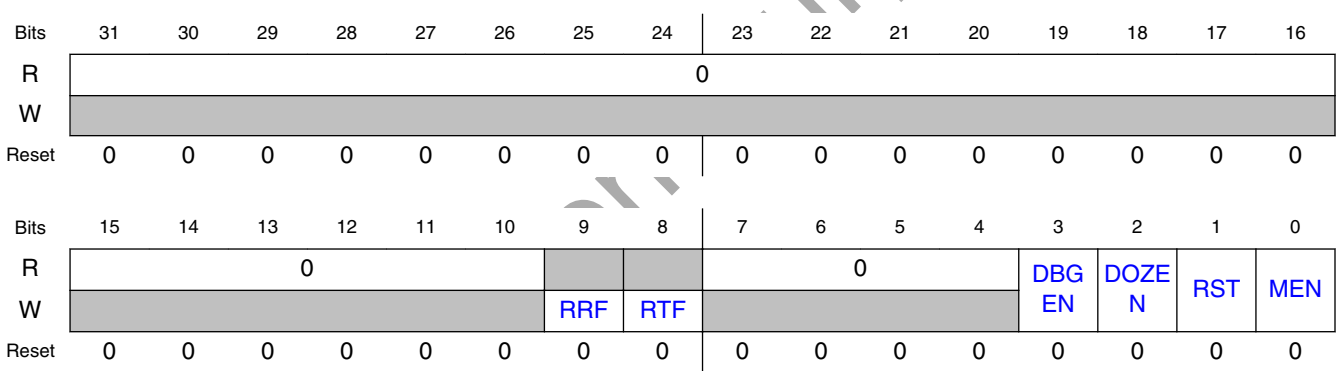
Field	Function
RXFIFO	The number of words in the receive FIFO is 2^{RXFIFO}.
7-0 TXFIFO	Transmit FIFO Size The number of words in the transmit FIFO is 2^{TXFIFO}.

42.3.1.4 Control (CR)

42.3.1.4.1 Address

Register	Offset
CR	10h

42.3.1.4.2 Diagram



42.3.1.4.3 Fields

Field	Function
31-10 —	Reserved
9 RRF	Reset Receive FIFO 0b - No effect. 1b - Receive FIFO is reset.
8 RTF	Reset Transmit FIFO 0b - No effect. 1b - Transmit FIFO is reset.
7-4 —	Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
3 DBGEN	Debug Enable 0b - Module is disabled in debug mode. 1b - Module is enabled in debug mode.
2 DOZEN	Doze mode enable Enables or disables Doze mode 0b - Module is enabled in Doze mode. 1b - Module is disabled in Doze mode.
1 RST	Software Reset Reset all internal logic and registers, except the Control Register. Remains set until cleared by software. 0b - Master logic is not reset. 1b - Master logic is reset.
0 MEN	Module Enable 0b - Module is disabled. 1b - Module is enabled.

42.3.1.5 Status (SR)

42.3.1.5.1 Address

Register	Offset
SR	14h

42.3.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							MBF	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		DMF	REF	TEF	TCF	FCF	WCF	0						RDF	TDF
W			w1c	w1c	w1c	w1c	w1c	w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.5.3 Fields

Field	Function
31-25 —	Reserved
24 MBF	Module Busy Flag 0b - LPSPI is idle. 1b - LPSPI is busy.
23-14 —	Reserved
13 DMF	Data Match Flag Indicates that the received data has matched the MATCH0 and/or MATCH1 fields as configured by MATCFG. 0b - Have not received matching data. 1b - Have received matching data.
12 REF	Receive Error Flag This flag will set when the Receiver FIFO overflows. 0b - Receive FIFO has not overflowed. 1b - Receive FIFO has overflowed.
11 TEF	Transmit Error Flag This flag will set when the Transmit FIFO underruns. 0b - Transmit FIFO underrun has not occurred. 1b - Transmit FIFO underrun has occurred
10 TCF	Transfer Complete Flag This flag will set in master mode when the LPSPI returns to idle state with the transmit FIFO empty. 0b - All transfers have not completed. 1b - All transfers have completed.
9 FCF	Frame Complete Flag This flag will set at the end of each frame transfer, when the PCS negates. 0b - Frame transfer has not completed. 1b - Frame transfer has completed.
8 WCF	Word Complete Flag This flag will set when the last bit of a received word is sampled. 0b - Transfer word not completed. 1b - Transfer word completed.
7-2 —	Reserved
1 RDF	Receive Data Flag The Receive Data Flag is set whenever the number of words in the receive FIFO is greater than RXWATER. 0b - Receive Data is not ready. 1b - Receive data is ready.
0 TDF	Transmit Data Flag The Transmit Data Flag is set whenever the number of words in the transmit FIFO is equal or less than TXWATER. 0b - Transmit data not requested. 1b - Transmit data is requested.

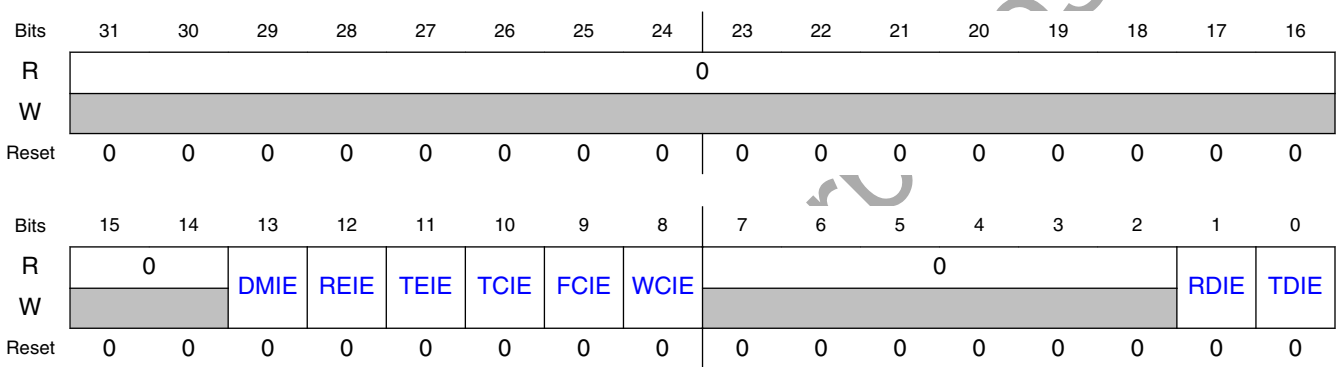
Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.6 Interrupt Enable (IER)

42.3.1.6.1 Address

Register	Offset
IER	18h

42.3.1.6.2 Diagram



42.3.1.6.3 Fields

Field	Function
31-14 —	Reserved
13 DMIE	Data Match Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
12 REIE	Receive Error Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
11 TEIE	Transmit Error Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
10 TCIE	Transfer Complete Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
9 FCIE	Frame Complete Interrupt Enable 0b - Interrupt disabled.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	1b - Interrupt enabled.
8 WCIE	Word Complete Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
7-2 —	Reserved
1 RDIE	Receive Data Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
0 TDIE	Transmit Data Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled

42.3.1.7 DMA Enable (DER)

42.3.1.7.1 Address

Register	Offset
DER	1Ch

42.3.1.7.2 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														RDDE	TDDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.3.1.7.3 Fields

Field	Function
31-2	Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

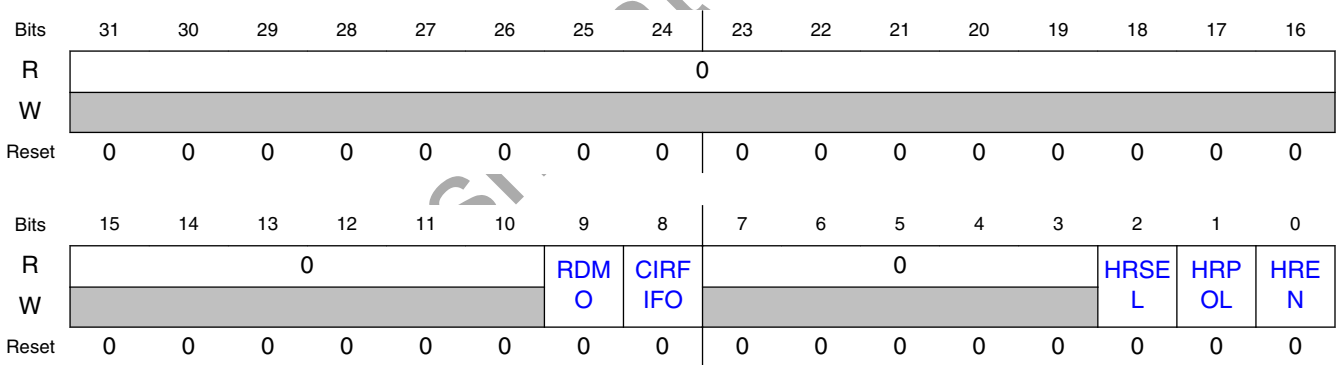
Field	Function
—	
1 RDDE	Receive Data DMA Enable 0b - DMA request disabled. 1b - DMA request enabled.
0 TDDE	Transmit Data DMA Enable 0b - DMA request disabled. 1b - DMA request enabled

42.3.1.8 Configuration 0 (CFGR0)

42.3.1.8.1 Address

Register	Offset
CFGR0	20h

42.3.1.8.2 Diagram



42.3.1.8.3 Fields

Field	Function
31-10 —	Reserved
9 RDMO	Receive Data Match Only When enabled, all received data that does not cause DMF to set is discarded. Once DMF is set, the RDMO configuration is ignored. When disabling RDMO, clear RDMO before clearing DMF to ensure no receive data is lost. 0b - Received data is stored in the receive FIFO as normal.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	1b - Received data is discarded unless the DMF is set.
8 CIRFIFO	Circular FIFO Enable When enabled, the transmit FIFO read pointer is saved to a temporary register. The transmit FIFO will be emptied as normal, but once the LPSPI is idle and the transmit FIFO is empty, then the read pointer value will be restored from the temporary register. This will cause the contents of the transmit FIFO to be cycled through repeatedly. 0b - Circular FIFO is disabled. 1b - Circular FIFO is enabled.
7-3 —	Reserved
2 HRSEL	Host Request Select Selects the source of the host request input. When the host request function is enabled with the LPSPI_HREQ pin, the LPSPI_PCS[1] function is disabled. 0b - Host request input is pin LPSPI_HREQ. 1b - Host request input is input trigger.
1 HRPOL	Host Request Polarity Configures the polarity of the host request pin. 0b - Active low. 1b - Active high.
0 HREN	Host Request Enable When enabled in master mode, the LPSPI will only initiate a SPI bus transfer if the host request input is asserted. 0b - Host request is disabled. 1b - Host request is enabled.

42.3.1.9 Configuration 1 (CFGR1)

42.3.1.9.1 Address

Register	Offset
CFGR1	24h

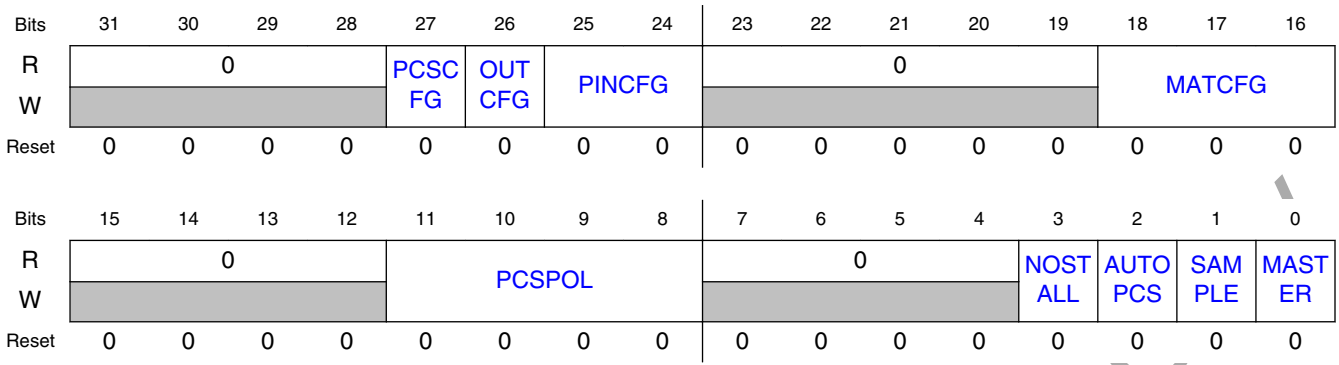
42.3.1.9.2 Function

The CFGR1 should only be written when the LPSPI is disabled.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.9.3 Diagram



42.3.1.9.4 Fields

Field	Function
31-28 —	Reserved
27 PCSCFG	Peripheral Chip Select Configuration PCSCFG must be set if performing 4-bit transfers. 0b - PCS[3:2] are enabled. 1b - PCS[3:2] are disabled.
26 OUTCFG	Output Config Configures if the output data is tristated between accesses (LPSPI_PCS is negated). 0b - Output data retains last value when chip select is negated. 1b - Output data is tristated when chip select is negated.
25-24 PINCFG	Pin Configuration Configures which pins are used for input and output data during single bit transfers. 00b - SIN is used for input data and SOUT for output data. 01b - SIN is used for both input and output data. 10b - SOUT is used for both input and output data. 11b - SOUT is used for input data and SIN for output data.
23-19 —	Reserved
18-16 MATCFG	Match Configuration Configures the condition that will cause the DMF to set. 000b - Match disabled. 001b - Reserved 010b - Match enabled (1st data word equals MATCH0 OR MATCH1). 011b - Match enabled (any data word equals MATCH0 OR MATCH1). 100b - Match enabled (1st data word equals MATCH0 AND 2nd data word equals MATCH1). 101b - Match enabled (any data word equals MATCH0 AND next data word equals MATCH1) 110b - Match enabled (1st data word AND MATCH1 equals MATCH0 AND MATCH1) 111b - Match enabled (any data word AND MATCH1 equals MATCH0 AND MATCH1).
15-12 —	Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
11-8 PCSPOL	Peripheral Chip Select Polarity Configures the polarity of each Peripheral Chip Select pin. 0000b - The PCSx is active low. 0001b - The PCSx is active high.
7-4 —	Reserved
3 NOSTALL	No Stall In master mode, the LPSPI will stall transfers when the transmit FIFO is empty or receive FIFO is full ensuring that no transmit FIFO underrun or receive FIFO overrun can occur. Setting this bit will disable this functionality. 0b - Transfers will stall when transmit FIFO is empty or receive FIFO is full. 1b - Transfers will not stall, allowing transmit FIFO underrun or receive FIFO overrun to occur.
2 AUTOPCS	Automatic PCS The LPSPI slave normally requires the PCS to negate between frames for correct operation. Setting this bit will cause the LPSPI to generate an internal PCS signal at the end of each transfer word when CPHA=1. When this bit is set, the SCK must remain idle for at least 4 LPSPI functional clock cycles (divided by PRESCALE configuration) between each word to ensure correct operation. This bit is ignored in master mode. 0b - Automatic PCS generation disabled. 1b - Automatic PCS generation enabled.
1 SAMPLE	Sample Point When set, the LPSPI master will sample the input data on a delayed LPSPI_SCK edge. This improves the setup time when sampling data. The input data setup time in master mode with delayed LPSPI_SCK edge is equal to the input data setup time in slave mode. This bit is ignored in slave mode. 0b - Input data sampled on SCK edge. 1b - Input data sampled on delayed SCK edge.
0 MASTER	Master Mode Configures the LPSPI in master or slave mode. This bit directly controls the direction of the LPSPI_SCK and LPCPI_PCS pins. 0b - Slave mode. 1b - Master mode.

42.3.1.10 Data Match 0 (DMR0)

42.3.1.10.1 Address

Register	Offset
DMR0	30h

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH0															
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	MATCH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.3.1.10.3 Fields

Field	Function
31-0 MATCH0	Match 0 Value Compared against the received data when receive data match is enabled.

42.3.1.11 Data Match 1 (DMR1)

42.3.1.11.1 Address

Register	Offset
DMR1	34h

42.3.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH1															
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	MATCH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.11.3 Fields

Field	Function
31-0 MATCH1	Match 1 Value Compared against the received data when receive data match is enabled.

42.3.1.12 Clock Configuration (CCR)

42.3.1.12.1 Address

Register	Offset
CCR	40h

42.3.1.12.2 Function

The CCR is only used in master mode and cannot be changed when the LPSPI is enabled.

42.3.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCKPCS								PCSSCK							
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	DBT								SCKDIV							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

42.3.1.12.4 Fields

Field	Function
31-24 SCKPCS	SCK to PCS Delay Configures the delay in master mode from the last SCK edge to the PCS negation. The delay is equal to (SCKPCS + 1) cycles of the LPSPI functional clock divided by the PRESCALE configuration, and the minimum delay is 1 cycle.
23-16	PCS to SCK Delay

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
PCSSCK	Configures the delay in master mode from the PCS assertion to the first SCK edge. The delay is equal to (PCSSCK + 1) cycles of the LPSPI functional clock divided by the PRESCALE configuration, and the minimum delay is 1 cycle.
15-8 DBT	Delay Between Transfers Configures the delay in master mode from the PCS negation to the next PCS assertion. The delay is equal to (DBT + 2) cycles of the LPSPI functional clock divided by the PRESCALE configuration, and the minimum delay is 2 cycles. Note that half the delay occurs before PCS assertion and the other half occurs after PCS negation, the full command word can only update in the middle. Also configures the delay in master mode from the last SCK edge of a transfer word and the first SCK edge of the next transfer word in a continuous transfer. The delay is equal to (DBT + 1) cycles of the LPSPI functional clock divided by the PRESCALE configuration, and the minimum delay is 1 cycle.
7-0 SCKDIV	SCK Divider Configures the divide ratio of the SCK pin in master mode. The SCK period is equal to (SCKDIV+2) cycles of the LPSPI functional clock divided by the PRESCALE configuration, and the minimum period is 2 cycles. If the period is an odd number of cycles, then the first half of the period will be one cycle longer than the second half.

42.3.1.13 FIFO Control (FCR)

42.3.1.13.1 Address

Register	Offset
FCR	58h

42.3.1.13.2 Diagram

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.13.3 Fields

Field	Function
31-24 —	Reserved
23-16 RXWATER	Receive FIFO Watermark The Receive Data Flag is set whenever the number of words in the receive FIFO is greater than RXWATER. Writing a value equal or greater than the FIFO size will be truncated.
15-8 —	Reserved
7-0 TXWATER	Transmit FIFO Watermark The Transmit Data Flag is set whenever the number of words in the transmit FIFO is equal or less than TXWATER. Writing a value equal or greater than the FIFO size will be truncated.

42.3.1.14 FIFO Status (FSR)

42.3.1.14.1 Address

Register	Offset
FSR	5Ch

42.3.1.14.2 Diagram

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

42.3.1.14.3 Fields

Field	Function
31-24	Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
—	
23-16 RXCOUNT	Receive FIFO Count Returns the number of words in the receive FIFO.
15-8 —	Reserved
7-0 TXCOUNT	Transmit FIFO Count Returns the number of words in the transmit FIFO.

42.3.1.15 Transmit Command (TCR)

42.3.1.15.1 Address

Register	Offset
TCR	60h

42.3.1.15.2 Function

Writes to either the Transmit Command Register or Transmit Data Register will push the data into the transmit FIFO, in the order that they are written. Command Register writes will be tagged and cause the command register to update, after that entry reaches the top of the FIFO. This allows changes to the command word and the transmit data itself to be interleaved. Changing the command word will cause all subsequent SPI bus transfers to be performed using the new command word.

- **In master mode,** writing a new command word does not initiate a new transfer, unless TXMSK is set. Transfers are initiated by transmit data in the transmit FIFO, or by a new command word (with TXMSK set). Hardware will clear TXMSK when the LPSPI_PCS negates.
- **In master mode,** if the command word is changed before an existing frame has completed, then the existing frame will terminate and the command word will then update. The command word can be changed during a continuous transfer, if CONTC of the new command word is set and the command word is written on a frame size boundary.
- **In slave mode,** the command word should be changed only when the LPSPI is idle and there is no SPI bus transfer.

Reading the Transmit Command Register will return the current state of the command register.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CPOL	CPHA	PRESCALE			0	PCS		LSBF	BYSW	CONT	CONTC	RXMSK	TXMSK	WIDTH	
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				FRAMESZ											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

42.3.1.15.4 Fields

Field	Function
31 CPOL	Clock Polarity This field is only updated between frames. 0b - The inactive state value of SCK is low. 1b - The inactive state value of SCK is high.
30 CPHA	Clock Phase This field is only updated between frames. 0b - Data is captured on the leading edge of SCK and changed on the following edge. 1b - Data is changed on the leading edge of SCK and captured on the following edge.
29-27 PRESCALE	Prescaler Value Prescaler applied to the clock configuration register for all SPI bus transfers. This field is only updated between frames. 000b - Divide by 1. 001b - Divide by 2. 010b - Divide by 4. 011b - Divide by 8. 100b - Divide by 16. 101b - Divide by 32. 110b - Divide by 64. 111b - Divide by 128.
26 —	Reserved
25-24 PCS	Peripheral Chip Select Configures the peripheral chip select used for the transfer. This field is only updated between frames. 00b - Transfer using LPSPI_PCS[0] 01b - Transfer using LPSPI_PCS[1] 10b - Transfer using LPSPI_PCS[2] 11b - Transfer using LPSPI_PCS[3]
23 LSBF	LSB First 0b - Data is transferred MSB first. 1b - Data is transferred LSB first.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
22 BYSW	Byte Swap Byte swap will swap the contents of [31:24] with [7:0] and [23:16] with [15:8] for each transmit data word read from the FIFO and each received data word stored to the FIFO (or compared with match registers). 0b - Byte swap disabled. 1b - Byte swap enabled.
21 CONT	Continuous Transfer In master mode, continuous transfer will keep the PCS asserted at the end of the frame size, until a command word is received that starts a new frame. In slave mode, when continuous transfer is enabled the LPSPI will only transmit the first FRAMESZ bits, after which it will transmit received data assuming a 32-bit shift register. 0b - Continuous transfer disabled. 1b - Continuous transfer enabled.
20 CONTC	Continuing Command In master mode, this bit allows the command word to be changed within a continuous transfer. The initial command word must enable continuous transfer (CONT=1), the continuing command must set this bit (CONTC=1) and the continuing command word must be loaded on a frame size boundary. For example, if the continuous transfer has a frame size of 64-bits, then a continuing command word must be loaded on a 64-bit boundary. 0b - Command word for start of new transfer. 1b - Command word for continuing transfer.
19 RXMSK	Receive Data Mask When set, receive data is masked (receive data is not stored in receive FIFO). 0b - Normal transfer. 1b - Receive data is masked.
18 TXMSK	Transmit Data Mask When set, transmit data is masked (no data is loaded from transmit FIFO and output pin is tristated). In master mode, this bit will initiate a new transfer which cannot be aborted by another command word and the bit will be cleared by hardware at the end of the transfer. 0b - Normal transfer. 1b - Mask transmit data.
17-16 WIDTH	Transfer Width Either RXMSK or TXMSK must be set for 2-bit or 4-bit transfers. 00b - Single bit transfer. 01b - Two bit transfer. 10b - Four bit transfer. 11b - Reserved.
15-12 —	Reserved
11-0 FRAMESZ	Frame Size Configures the frame size in number of bits equal to (FRAMESZ + 1). The minimum frame size is 8 bits. If the frame size is larger than 32 bits, data will be loaded from the transmit FIFO and stored to the receive FIFO every 32 bits. If the size of the transfer word is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO will contain the remainder bits (e.g.: a 72-bit transfer will load/store 32-bits from the FIFO and then another 32-bits from the FIFO and then the final 8-bits from the FIFO).

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.16 Transmit Data (TDR)

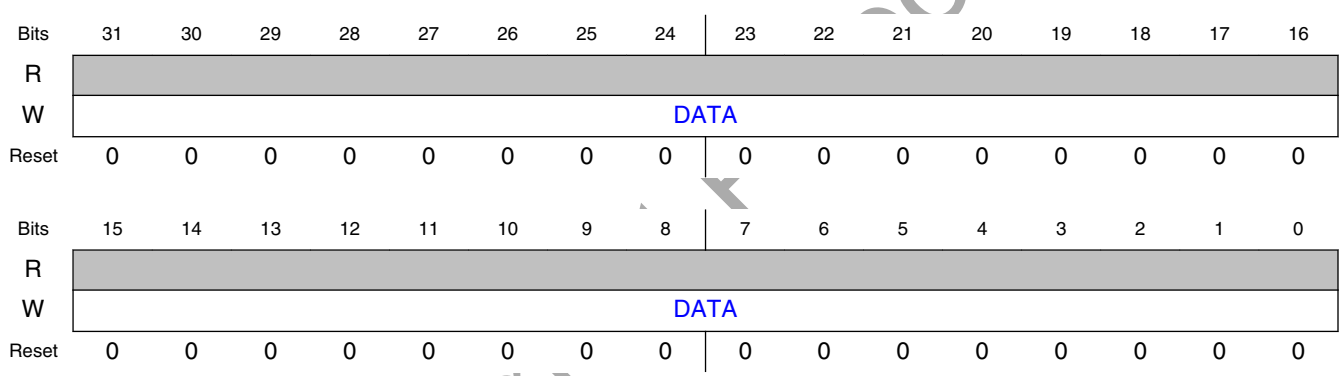
42.3.1.16.1 Address

Register	Offset
TDR	64h

42.3.1.16.2 Function

Writes to either the Transmit Command Register or Transmit Data Register will push the data into the transmit FIFO, in the order that it was written.

42.3.1.16.3 Diagram



42.3.1.16.4 Fields

Field	Function
31-0 DATA	Transmit Data Both 8-bit and 16-bit writes of transmit data will zero extend the data written and push the data into the transmit FIFO.

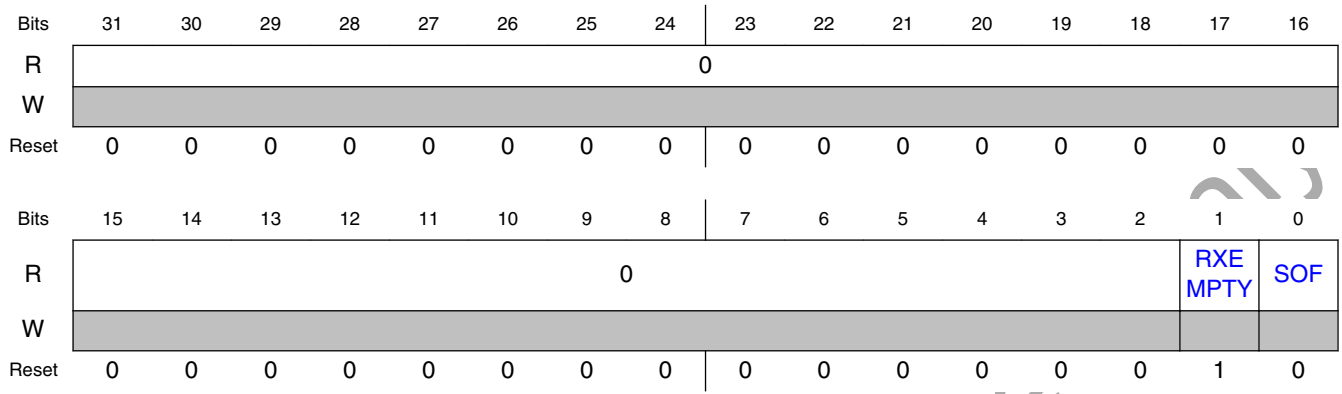
42.3.1.17 Receive Status (RSR)

42.3.1.17.1 Address

Register	Offset
RSR	70h

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.17.2 Diagram



42.3.1.17.3 Fields

Field	Function
31-2 —	Reserved
1 RXEMPTY	RX FIFO Empty 0b - RX FIFO is not empty. 1b - RX FIFO is empty.
0 SOF	Start Of Frame Indicates that this is the first data word received after LPSPI_PCS assertion. 0b - Subsequent data word received after LPSPI_PCS assertion. 1b - First data word received after LPSPI_PCS assertion.

42.3.1.18 Receive Data (RDR)

42.3.1.18.1 Address

Register	Offset
RDR	74h

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.3.1.18.2 Diagram

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

42.3.1.18.3 Fields

Field	Function
31-0 DATA	Receive Data

42.4 Functional description

42.4.1 Clocking and Resets

42.4.1.1 Functional clock

The LPSPI functional clock is asynchronous to the bus clock and can remain enabled in low power modes to support SPI bus transfers in both master and slave modes. If the functional clock is disabled in slave mode, the LPSPI can transfer a single word before the functional clock needs to be enabled. The LPSPI divides the functional clock by a prescaler and the resulting frequency must be at least two times faster than the SPI clock frequency.

42.4.1.2 External clock

The LPSPI shift register is clocked directly by the external pins. This allows the LPISPI slave to remain operational in low power modes, even when the LPSPI functional clock is disabled, although this is limited to a single word transfer.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.4.1.3 Bus clock

The bus clock is only used for bus accesses to the control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the LPSPI registers, including FIFOs.

42.4.1.4 Chip reset

The logic and registers for the LPSPI are reset to their default state on a chip reset.

42.4.1.5 Software reset

The LPSPI implements a software reset bit in the Control Register. The CR[RST] will reset all logic and registers to their default state, except for the CR itself.

42.4.1.6 FIFO reset

The LPSPI implements write-only control bits that resets the transmit/command FIFO (CR[RTF] and receive FIFO (CR[RRF]). A FIFO is empty after being reset.

42.4.2 Master Mode

42.4.2.1 Transmit and Command FIFO

The transmit and command FIFO is a combined FIFO that includes both transmit data and command words. Command words are stored to the transmit/command FIFO by writing the transmit command register. Transmit data words are stored to the transmit/command FIFO by writing the transmit data register.

When a command word is at the top of the transmit/command FIFO, the following actions can occur:

- If the LPSPI is between frames, the command word is pulled from the FIFO and controls all subsequent transfers.
- If the LPSPI is busy and either the existing CONT bit is clear or the new CONTC value is clear, the SPI frame will complete at the end of the existing word, ignoring

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

the FRAMESZ configuration. The command word is then pulled from the FIFO and controls all subsequent transfers (or until the next update to the command word).

- If the LPSPI is busy and the existing CONT bit is set and the new CONTC value is set, the command word is pulled from the FIFO during the last LPSPI_SCK pulse of the existing frame (based on FRAMESZ configuration) and the frame continues using the new command value for the rest of the frame (or until the next update to the command word). When CONTC is set, only the lower 24-bits of the command word are updated.

The current state of the existing command word can be read by reading the transmit command register. It requires at least three LPSPI functional clock cycles for the transmit command register to update after it is written (assuming an empty FIFO) and the LPSPI must be enabled (CR[MEN] is set).

Writing the transmit command register does not initiate a SPI bus transfer, unless the TXMSK bit is set. When TXMSK is set, a new command word will not be loaded until the end of the existing frame (based on FRAMESZ configuration) and the TXMSK bit will be cleared at the end of the transfer.

The following table describes the attributes that are controlled by the command word.

Purposes Only

Table 42-3. LPSPI Command Word

Field	Description	Modify During Transfer
CPOL	Configures polarity of the LPSPI_SCK pin. Any change of CPOL value will cause a transition on the LPSPI_SCK pin.	N
CPHA	Configures clock phase of transfer.	N
PRESCALE	Configures prescaler used to divide the LPSPI functional clock to generate the timing parameters of the SPI bus transfer. Changing PRESCALE in conjunction with PCS allows the LPSPI to connect to different slave devices at different frequencies.	N
PCS	Configures which LPSPI_PCS asserts for the transfer, the polarity of LPSPI_PCS is static and configured by PCSPOL. If PCSCFG is set, then PCS[3:2] should not be selected.	N
LSBF	Configures if LSB (bit 0) or MSB (bit 31 for a 32-bit word) is transmitted/received first.	Y
BYSW	Enables byte swap on each 32-bit word when transmitting and receiving data. Can be useful when interfacing to devices that organize data as big endean.	Y

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 42-3. LPSPI Command Word (continued)

Field	Description	Modify During Transfer
CONT	Configures for a continuous transfer that keeps PCS asserted between frames (as configured by FRAMESZ). A new command word is required to cause PCS to negate. Also supports changing the command word at frame size boundaries.	Y
CONTC	Indicates this is a new command word for the existing continuous transfer. This bit is ignored when not written to the transmit/command FIFO on a frame boundary.	Y
RXMSK	Masks the receive data and does not store to the receive FIFO or perform receive data matching. Useful for half-duplex transfers or to configure which fields are compared during receive data matching.	Y
TXMSK	Masks the transmit data, so that data is not pulled from transmit FIFO and the output data pin is tristated (unless configured by OUTCFG). Useful for half-duplex transfers.	Y
WIDTH	Configures the number of bits shifted on each LPSPI_SCK pulse. Single bit transfers support traditional SPI bus transfers in either half-duplex or full-duplex data formats. Two and four bit transfers are useful for interfacing to QuadSPI memory devices and only support half-duplex data formats (at least one of TXMSK or RXMSK must also be set).	Y
FRAMESZ	Configures the number of bits in each frame to FRAMESZ+1. The minimum frame size is 8-bits and the maximum frame size is 4096-bits. If the frame size is less than or equal to 32-bits, the word size and frame size are identical. If the frame size is greater than 32-bits, then the word size is 32-bits for each word except the last (the last word contains the remainder bits if the frame size is not divisible by 32). The minimum word size is 2-bits, a frame size of 33-bits (or similar) is not supported.	Y

The LPSPI initiates a SPI bus transfer when data is written to the transmit FIFO, the HREQ pin is asserted (or disabled) and the LPSPI is enabled. The SPI bus transfer uses the attributes configured in the transmit command register and timing parameters from the clock configuration register to perform the transfer. The SPI bus transfer ends once

Freescale Confidential Proprietary - Non-Disclosure Agreement required

the FRAMESZ configuration is reached, or at the end of a word when a new transmit command word is at the top of the transmit/command FIFO. The HREQ input is only checked the next time the LPSPI goes idle (completes the current transfer and transmit/command register is empty).

The transmit/command FIFO also supports a Circular FIFO feature. This allows the LPSPI master to (periodically) repeat a short data transfer that can fit within the transmit/command FIFO, without requiring additional FIFO accesses. When the circular FIFO is enabled, the current state of the FIFO read pointer is saved and the status flags do not update. Once the transmit/command FIFO is considered empty and the LPSPI is idle, the FIFO read pointer is restored with the saved version, so the contents of the transmit/command FIFO are not permanently pulled from the FIFO while circular FIFO mode is enabled.

42.4.2.2 Receive FIFO and Data Match

The receive FIFO is used to store receive data during SPI bus transfers. When RXMSK is set, receive data is discarded instead of storing in the receive FIFO.

Receive data supports a receive data match function that can match received data against one of two words or against a masked data word. The data match function can also be configured to compare only the first one or two received data words since the start of the frame. Receive data that is already discarded due to RXMSK bit cannot cause the data match to set and will delay the match on first received data word until after all discarded data is received. The receiver match function can also be configured to discard all receive data until a data match is detected, using the CFGR0[RDMO] control bit. When clearing the CFGR0[RDMO] control bit following a data match, clear CFGR0[RDMO] before clearing SR[DMF] to allow all subsequent data to be received.

42.4.2.3 Timing Parameters

The following table lists the timing parameters that are used for all SPI bus transfers, these timing parameters are relative to the LPSPI functional clock divided by the PRESCALE configuration. Although the Clock Configuration Register cannot be changed when the LPSPI is busy, the PRESCALE configuration can be altered between transfers using the command register, to support interfacing to different slave devices at different frequencies.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 42-4. LPSPI Timing Parameters

Field	Description	Min	Max
SCKDIV	Configures the LPSPI_SCK clock period to (SCKDIV+2) cycles. When configured to an odd number of cycles, the first half of the LPSPI_SCK cycle is one cycle longer than the second half.	0 (2 cycles)	255 (257 cycles)
DBT	Configures the minimum delay between PCS negation and the next PCS assertion to (DBT + 2) cycles. When the command word is updated between transfers, there is a minimum of (DBT/2)+1 cycles between the command word update and any change on LPSPI_PCS pins.	0 (2 cycles)	255 (257 cycles)
DBT	Configures the delay during a continuous transfer between the last SCK edge of a frame and the first SCK edge of the continuing frame to (DBT + 1) cycles. This is useful where the external slave requires a large delay between different words of a SPI bus transfer.	0 (1 cycle)	255 (256 cycles)
PCSSCK	Configures the minimum delay between PCS assertion and the first SCK edge to (PCSSCK + 1) cycles.	0 (1 cycle)	255 (256 cycles)
SCKPCS	Configures the minimum delay between the last SCK edge and the PCS assertion to (SCKPCS + 1) cycles.	0 (1 cycle)	255 (256 cycles)

For Asse

42.4.2.4 Pin Configuration

The LPSPI_SIN and LPSPI_SOUT pins can be configured via the PINCFG configuration to swap directions or even support half-duplex transfers on the same pin.

The OUTCFG configuration can be used to determine if output data pin (eg: LPSPI_SOUT) will tristate when the LPSPI_PCS is negated, or if it will simply retain the last value. When configuring for half-duplex transfers using the same data pin in single bit transfer mode, or any transfer in 2-bit and 4-bit transfer modes, then the output data pins must be configured to tristate when LPSPI_PCS is negated.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The PCSCFG configuration is used to disable LPSPI_PCS[3:2] functions and to use them for quad-data transfers. This option must be enabled when performing quad-data transfers.

42.4.3 Slave Mode

LPSPI slave mode uses the same shift register and logic as the master mode, but does not use the clock configuration register and the transmit command register must remain static during SPI bus transfers.

42.4.3.1 Transmit and Command FIFO

The transmit command register should be initialized before enabling the LPSPI in slave mode, although the command register will not update until after the LPSPI is enabled. Once enabled, the transmit command register should only be changed if the LPSPI is idle. The following table lists how the command register functions in slave mode.

Table 42-5. LPSPI Command Word in Slave Mode

Field	Description
CPOL	Configures polarity of the external LPSPI_SCK input.
CPHA	Configures clock phase of transfer.
PRESCALE	Configures LPSPI functional clock prescaler.
PCS	Configures which LPSPI_PCS is used, the polarity of LPSPI_PCS is static and configured by PCSPOL. If PCSCFG is set, then PCS[3:2] should not be selected.
LSBF	Configures if LSB (bit 0) or MSB (bit 31 for a 32-bit word) is transmitted/received first.
BYSW	Enables byte swap on each 32-bit word when transmitting and receiving data. Can be useful when interfacing to devices that organize data as big endean.
CONT	When set, only the first FRAMSZ bits will be transmitted/ received by the LPSPI.
CONTC	This bit is reserved in slave mode.
RXMSK	Masks the receive data and does not store to the receive FIFO or perform receive data matching. Useful for half-duplex transfers or to configure which fields are compared during receive data matching.
TXMSK	Masks the transmit data, so that data is not pulled from transmit FIFO and the output data pin is tristated (unless configured by OUTCFG). Useful for half-duplex transfers.
WIDTH	Configures the number of bits shifted on each LPSPI_SCK pulse. Single bit transfers support traditional SPI bus transfers in either half-duplex or full-duplex data formats. Two and four bit transfers are useful for interfacing to QuadSPI memory

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 42-5. LPSPI Command Word in Slave Mode (continued)

Field	Description
	devices and only support half-duplex data formats (at least one of TXMSK or RXMSK must also be set).
FRAMESZ	Configures the number of bits in each frame to FRAMESZ+1. The minimum frame size is 8-bits and the maximum frame size is 4096-bits. If the frame size is less than or equal to 32-bits, the word size and frame size are identical. If the frame size is greater than 32-bits, then the word size is 32-bits for each word except the last (the last word contains the remainder bits if the frame size is not divisible by 32). The minimum word size is 2-bits, a frame size of 33-bits (or similar) is not supported.

The transmit FIFO must be filled with transmit data before the LPSPI_PCS input asserts, otherwise the transmit error flag will set.

42.4.3.2 Receive FIFO and Data Match

The receive FIFO is used to store receive data during SPI bus transfers. When RXMSK is set, receive data is discarded instead of storing in the receive FIFO.

Receive data supports a receive data match function that can match received data against one of two words or against a masked data word. The data match function can also be configured to compare only the first one or two received data words since the start of the frame. Receive data that is already discarded due to RXMSK bit cannot cause the data match to set and will delay the match on first received data word until after all discarded data is received. The receiver match function can also be configured to discard all receive data until a data match is detected, using the CFGR0[RDMO] control bit. When clearing the CFGR0[RDMO] control bit following a data match, clear CFGR0[RDMO] before clearing SR[DMF] to allow all subsequent data to be received.

42.4.3.3 Clocked Interface

The LPSPI supports interfacing to external masters that provide only clock and data pins (LPSPI_PCS is not required). This requires using CPHA=1, configuring the LPSPI_PCS input to be always asserted (configure PCSPOL) and setting the AUTOPCS bit. When AUTOPCS is set, a minimum of 4 LPSPI functional clock cycles (divided by PRESCALE configuration) is required between the last LPSPI_SCK edge of one word and the first LPSPI_SCK edge of the next word.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.4.4 Interrupts and DMA Requests

The following table illustrates the status flags that can generate the LPSPI interrupt and LPSPI transmit/receive DMA requests.

Table 42-6. LPSPI Interrupts and DMA Requests

Flag	Description	Interrupt	DMA Request	Low Power Wakeup
TDF	Data can be written to transmit FIFO, as configured by TXWATER.	Y	TX	Y
RDF	Data can be read from the receive FIFO, as configured by RXWATER.	Y	RX	Y
WCF	Word complete, last bit of word has been sampled.	Y	N	Y
FCF	Frame complete, PCS has negated .	Y	N	Y
TCF	Transfer complete, PCS has negated and transmit/command FIFO is empty.	Y	N	Y
TEF	Transmit error flag, indicates transmit/ command FIFO underrun. This bit cannot set in master mode when NOSTALL is clear.	Y	N	Y
REF	Receive error flag, indicates receive FIFO overflow. This bit cannot set in master mode when NOSTALL is clear.	Y	N	Y
DMF	Data match flag, received data has matched the configured data match value.	Y	N	Y
MBF	LPSPI is busy performing a SPI bus transfer.	N	N	N

42.4.5 Peripheral Triggers

The connection of the LPSPI peripheral triggers with other peripherals are device specific.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

42.4.5.1 Output Triggers

The LPSPI generates two output triggers that can be connected to other peripherals on the device. The frame output trigger asserts at the end of each frame (when PCS negates) and remains asserted until PCS next asserts. The word output trigger asserts at the end of each received word and remains asserted for one LPSPI_SCK period.

42.4.5.2 Input Trigger

The LPSPI input trigger can be selected in place of the LPSPI_HREQ input to control the start of a LPSPI bus transfer. The input trigger must assert for longer than one LPSPI functional clock cycle to be detected.

For Assessment Purposes Only

Chapter 43
Low Power Inter-Integrated Circuit (LPI2C)

43.1 Chip-specific Low Power Inter-Integrated Circuit (LPI2C) information

43.1.1 Instantiation Information

This device has one LPI2C module.

The LPI2C module includes SMBus support and DMA support. It also has optional address match wakeup in Stop/VLPS mode.

Table 43-1. LPI2C Configuration

LPI2C Feature	LPI2C0
TX FIFO (word)	4
RX FIFO (word)	4
SMBus	Yes
Slave mode enable	Yes

NOTE

Low leakage mode is not supported in this device.

43.2 Introduction

43.2.1 Overview

The LPI2C is a low power Inter-Integrated Circuit (I2C) module that supports an efficient interface to an I2C bus as a master and/or a slave. The LPI2C can continue operating in stop modes provided an appropriate clock is available and is designed for low CPU

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

overhead with DMA offloading of FIFO register accesses. The LPI2C implements logic support for standard-mode, fast-mode, fast-mode plus and ultra-fast modes of operation. The LPI2C module also complies with the System Management Bus (SMBus) Specification, version 2.

43.2.2 Features

The LPI2C supports the following features of the I2C specification:

- Standard, Fast, Fast+ and Ultra Fast modes are supported.
- HS-mode supported in slave mode.
- HS-mode supported for master mode, provided SCL pin implements current source pull-up (device specific).
- Multi-master support including synchronization and arbitration.
- Clock stretching.
- General call, 7-bit and 10-bit addressing.
- Software reset, START byte and Device ID require software support.

The LPI2C master supports the following features:

- Command/transmit FIFO of 4words.
- Receive FIFO of 4words.
- Command FIFO will wait for idle I2C bus before initiating transfer
- Command FIFO can initiate (repeated) START and STOP conditions and one or more master-receiver transfers.
- STOP condition can be generated from command FIFO or automatically when the transmit FIFO is empty.
- Host request input can be used to control the start time of an I2C bus transfer.
- Flexible receive data match can generate interrupt on data match and/or discard unwanted data.
- Flag and optional interrupt to signal Repeated START condition, STOP condition, loss of arbitration, unexpected NACK and command word errors.
- Supports configurable bus idle timeout and pin stuck low timeout.

The LPI2C slave supports the following features:

- Separate I2C slave registers to minimize software overhead due to master/slave switching.
- Support for 7-bit or 10-bit addressing, address range, SMBus alert and general call address.
- Transmit data register supporting interrupt or DMA requests.
- Receive data register supporting interrupt or DMA requests.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- Software controllable ACK or NACK, with optional clock stretching on ACK/ NACK bit.
- Configurable clock stretching to avoid transmit FIFO underrun and receive FIFO overrun.
- Flag and optional interrupt at end of packet, STOP condition or bit error detection.

43.2.3 Block Diagram

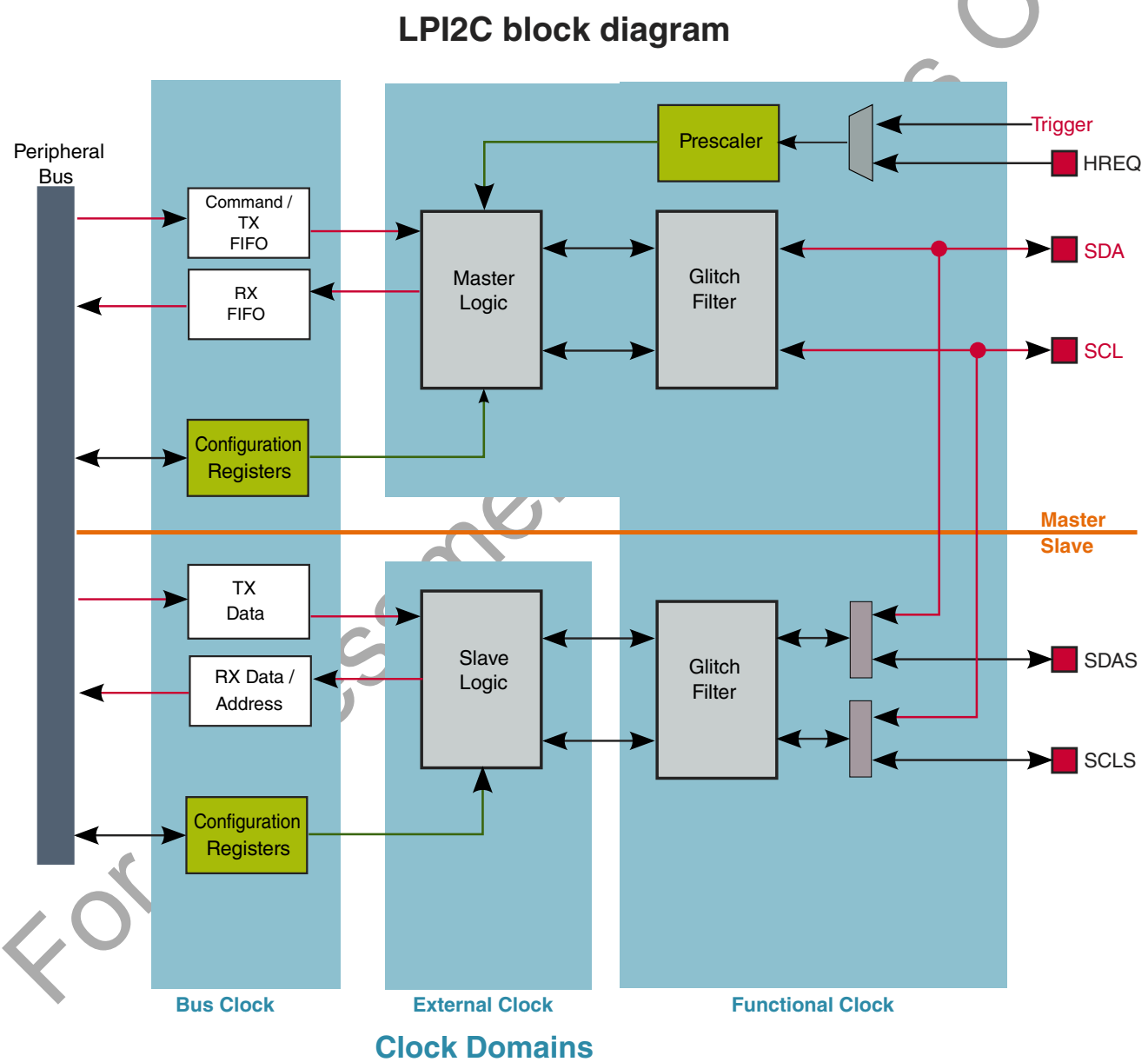


Figure 43-1. LPI2C block diagram

For Assessment Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.2.4 Modes of operation

The LPI2C module supports the chip modes described in the following table.

Table 43-2. Chip modes supported by the LPI2C module

Chip mode	LPI2C Operation
Run	Normal operation
Stop/Wait	Can continue operating provided the Doze Enable bit (MCR[DOZEN]) is clear and the LPI2C is using an external or internal clock source which remains operating during stop/wait modes.
Low Leakage Stop	The Doze Enable (MCR[DOZEN]) bit is ignored and the LPI2C will wait for the current transfer to complete any pending operation before acknowledging low leakage mode entry.
Debug	Can continue operating provided the Debug Enable bit (MCR[DBGE]) is set.

43.2.5 Signal Descriptions

Signal	Description	I/O
SCL	LPI2C clock line. In 4-wire mode, this is the SCL input pin.	I/O
SDA	LPI2C data line. In 4-wire mode, this is the SDA input pin.	I/O
HREQ	Host request, can initiate an LPI2C master transfer if asserted and the I2C bus is idle.	I
SCLS	Secondary I2C clock line. In 4-wire mode, this is the SCLS output pin. If LPI2C master/slave are configured to use separate pins, this the LPI2C slave SCL pin.	I/O
SDAS	Secondary I2C data line. In 4-wire mode, this is the SDAS output pin. If LPI2C master/slave are configured to use separate pins, this the LPI2C slave SDA pin.	I/O

For

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3 Memory Map and Registers

43.3.1 LPI2C Register Descriptions

Only

43.3.1.1 LPI2C Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID Register (VERID)	32	RO	01000003h
4h	Parameter Register (PARAM)	32	RO	00000202h
10h	Master Control Register (MCR)	32	RW	00000000h
14h	Master Status Register (MSR)	32	W1C	00000001h
18h	Master Interrupt Enable Register (MIER)	32	RW	00000000h
1Ch	Master DMA Enable Register (MDER)	32	RW	00000000h
20h	Master Configuration Register 0 (MCFGR0)	32	RW	00000000h
24h	Master Configuration Register 1 (MCFGR1)	32	RW	00000000h
28h	Master Configuration Register 2 (MCFGR2)	32	RW	00000000h
2Ch	Master Configuration Register 3 (MCFGR3)	32	RW	00000000h
40h	Master Data Match Register (MDMR)	32	RW	00000000h
48h	Master Clock Configuration Register 0 (MCCR0)	32	RW	00000000h
50h	Master Clock Configuration Register 1 (MCCR1)	32	RW	00000000h
58h	Master FIFO Control Register (MFCR)	32	RW	00000000h
5Ch	Master FIFO Status Register (MFSR)	32	RO	00000000h
60h	Master Transmit Data Register (MTDR)	32	WORZ	00000000h
70h	Master Receive Data Register (MRDR)	32	RO	00004000h
110h	Slave Control Register (SCR)	32	RW	00000000h
114h	Slave Status Register (SSR)	32	W1C	00000000h
118h	Slave Interrupt Enable Register (SIER)	32	RW	00000000h
11Ch	Slave DMA Enable Register (SDER)	32	RW	00000000h
124h	Slave Configuration Register 1 (SCFGR1)	32	RW	00000000h
128h	Slave Configuration Register 2 (SCFGR2)	32	RW	00000000h
140h	Slave Address Match Register (SAMR)	32	RW	00000000h
150h	Slave Address Status Register (SASR)	32	RO	00004000h
154h	Slave Transmit ACK Register (STAR)	32	RW	00000000h
160h	Slave Transmit Data Register (STDR)	32	WORZ	00000000h
170h	Slave Receive Data Register (SRDR)	32	RO	00004000h

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.2 Version ID Register (VERID)

43.3.1.2.1 Address

Register	Offset
VERID	0h

43.3.1.2.2 Function

.

43.3.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	0	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	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

43.3.1.2.4 Fields

Field	Function
31-24 MAJOR	Major Version Number This read only field returns the major version number for the specification.
23-16 MINOR	Minor Version Number This read only field returns the minor version number for the specification.
15-0 FEATURE	Feature Specification Number This read only field returns the feature set number. 0000000000000010b - Master only with standard feature set. 0000000000000011b - Master and slave with standard feature set.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.3 Parameter Register (PARAM)

43.3.1.3.1 Address

Register	Offset
PARAM	4h

43.3.1.3.2 Function

.

43.3.1.3.3 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				MRXFIFO				0				MTXFIFO			
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

43.3.1.3.4 Fields

Field	Function
31-16 —	Reserved
15-12 —	Reserved
11-8 MRXFIFO	Master Receive FIFO Size The number of words in the master receive FIFO is 2^MRXFIFO.
7-4 —	Reserved
3-0 MTXFIFO	Master Transmit FIFO Size The number of words in the master transmit FIFO is 2^MTXFIFO.

Freescale Confidential Proprietary Use Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.4 Master Control Register (MCR)

43.3.1.4.1 Address

Register	Offset
MCR	10h

43.3.1.4.2 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	0						0	0	0				DBGEN	DOZEN	RST	MEN
W							RRF	RTF								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.4.3 Fields

Field	Function
31-10 Reserved	Reserved
9 RRF	Reset Receive FIFO 0b - No effect. 1b - Receive FIFO is reset.
8 RTF	Reset Transmit FIFO 0b - No effect. 1b - Transmit FIFO is reset.
7-4 —	Reserved
3 DBGEN	Debug Enable 0b - Master is disabled in debug mode. 1b - Master is enabled in debug mode.
2 DOZEN	Doze mode enable Enables or disables Doze mode for the master. 0b - Master is enabled in Doze mode. 1b - Master is disabled in Doze mode.
1 RST	Software Reset

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	Reset all internal master logic and registers, except the Master Control Register. Remains set until cleared by software. 0b - Master logic is not reset. 1b - Master logic is reset.
0 MEN	Master Enable 0b - Master logic is disabled. 1b - Master logic is enabled.

43.3.1.5 Master Status Register (MSR)

43.3.1.5.1 Address

Register	Offset
MSR	14h

43.3.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BBF	MBF	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	DMF	PLTF	FEF	ALF	NDF	SDF	EPF	0						RDF	TDF
W		w1c	w1c	w1c	w1c	w1c	w1c	w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

43.3.1.5.3 Fields

Field	Function
31-26 —	Reserved
25 BBF	Bus Busy Flag 0b - I2C Bus is idle. 1b - I2C Bus is busy.
24 MBF	Master Busy Flag 0b - I2C Master is idle. 1b - I2C Master is busy.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
23-15 Reserved	Reserved
14 DMF	Data Match Flag Indicates that the received data has matched the MATCH0 and/or MATCH1 fields as configured by MATCFG. Received data that is discarded due to CMD field does not cause this flag to set. 0b - Have not received matching data. 1b - Have received matching data.
13 PLTF	Pin Low Timeout Flag Will set when the SCL and/or SDA input is low for more than PINLOW cycles, even when the LPI2C master is idle. Software is responsible for resolving the pin low condition. This flag cannot be cleared for as long as the pin low timeout continues and must be cleared before the LPI2C can initiate a START condition. 0b - Pin low timeout has not occurred or is disabled. 1b - Pin low timeout has occurred.
12 FEF	FIFO Error Flag Detects an attempt to send or receive data without first generating a (repeated) START condition. This can occur if the transmit FIFO underflows when the AUTOSTOP bit is set. When this flag is set, the LPI2C master will send a STOP condition (if busy) and will not initiate a new START condition until this flag has been cleared. 0b - No error. 1b - Master sending or receiving data without START condition.
11 ALF	Arbitration Lost Flag This flag will set if the LPI2C master transmits a logic one and detects a logic zero on the I2C bus, or if it detects a START or STOP condition while it is transmitting data. When this flag sets, the LPI2C master will release the bus (go idle) and will not initiate a new START condition until this flag has been cleared. 0b - Master has not lost arbitration. 1b - Master has lost arbitration.
10 NDF	NACK Detect Flag This flag will set if the LPI2C master detects a NACK when transmitting an address or data. If a NACK is expected for a given address (as configured by the command word) then the flag will set if a NACK is not generated. When set, the master will transmit a STOP condition and will not initiate a new START condition until this flag has been cleared. 0b - Unexpected NACK not detected. 1b - Unexpected NACK was detected.
9 SDF	STOP Detect Flag This flag will set when the LPI2C master generates a STOP condition. 0b - Master has not generated a STOP condition. 1b - Master has generated a STOP condition.
8 EPF	End Packet Flag This flag will set when the LPI2C master generates either a repeated START or a STOP condition. It does not set when the master first generates a START condition. 0b - Master has not generated a STOP or Repeated START condition. 1b - Master has generated a STOP or Repeated START condition.
7-2 —	Reserved
1 RDF	Receive Data Flag The Receive Data Flag is set whenever the number of words in the receive FIFO is greater than RXWATER.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0b - Receive Data is not ready. 1b - Receive data is ready.
0 TDF	Transmit Data Flag The Transmit Data Flag is set whenever the number of words in the transmit FIFO is equal or less than TXWATER. 0b - Transmit data not requested. 1b - Transmit data is requested.

43.3.1.6 Master Interrupt Enable Register (MIER)

43.3.1.6.1 Address

Register	Offset
MIER	18h

43.3.1.6.2 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	DMIE	PLTIE	FEIE	ALIE	NDIE	SDIE	EPIE	0						RDIE	TDIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.6.3 Fields

Field	Function
31-15 Reserved	Reserved
14 DMIE	Data Match Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
13 PLTIE	Pin Low Timeout Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

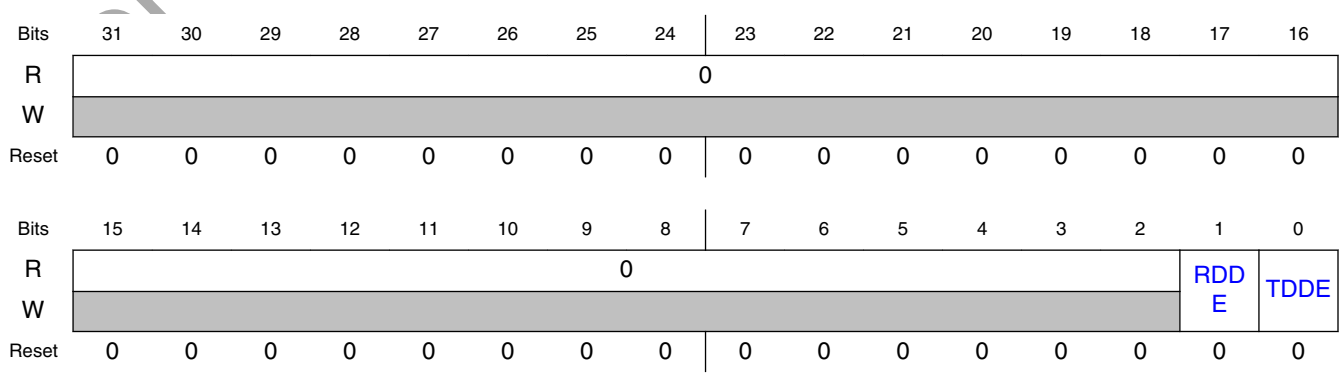
Field	Function
12 FEIE	FIFO Error Interrupt Enable 0b - Interrupt enabled. 1b - Interrupt disabled.
11 ALIE	Arbitration Lost Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
10 NDIE	NACK Detect Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
9 SDIE	STOP Detect Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
8 EPIE	End Packet Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
7-2 —	Reserved
1 RDIE	Receive Data Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
0 TDIE	Transmit Data Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled

43.3.1.7 Master DMA Enable Register (MDER)

43.3.1.7.1 Address

Register	Offset
MDER	1Ch

43.3.1.7.2 Diagram



Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.7.3 Fields

Field	Function
31-2 Reserved	Reserved
1 RDDE	Receive Data DMA Enable 0b - DMA request disabled. 1b - DMA request enabled.
0 TDDE	Transmit Data DMA Enable 0b - DMA request disabled. 1b - DMA request enabled

43.3.1.8 Master Configuration Register 0 (MCFGR0)

43.3.1.8.1 Address

Register	Offset
MCFGR0	20h

43.3.1.8.2 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						RDMO	CIRFIFO	0					HRSEL	HRPOL	HREN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.8.3 Fields

Field	Function
31-10 Reserved	Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
9 RDMO	Receive Data Match Only When enabled, all received data that does not cause DMF to set is discarded. Once DMF is set, the RDMO configuration is ignored. When disabling RDMO, clear RDMO before clearing DMF to ensure no receive data is lost. 0b - Received data is stored in the receive FIFO as normal. 1b - Received data is discarded unless the RMF is set.
8 CIRFIFO	Circular FIFO Enable When enabled, the transmit FIFO read pointer is saved to a temporary register. The transmit FIFO will be emptied as normal, but once the LPI2C master is idle and the transmit FIFO is empty, then the read pointer value will be restored from the temporary register. This will cause the contents of the transmit FIFO to be cycled through repeatedly. If AUTOSTOP is set, a STOP condition will be sent whenever the transmit FIFO is empty and the read pointer is restored. 0b - Circular FIFO is disabled. 1b - Circular FIFO is enabled.
7-3 —	Reserved
2 HRSEL	Host Request Select Selects the source of the host request input. 0b - Host request input is pin HREQ. 1b - Host request input is input trigger.
1 HRPOL	Host Request Polarity Configures the polarity of the host request input pin. 0b - Active low. 1b - Active high.
0 HREN	Host Request Enable When enabled, the LPI2C master will only initiate a START condition if the host request input is asserted and the bus is idle. A repeated START is not affected by the host request. 0b - Host request input is disabled. 1b - Host request input is enabled.

43.3.1.9 Master Configuration Register 1 (MCFGR1)

43.3.1.9.1 Address

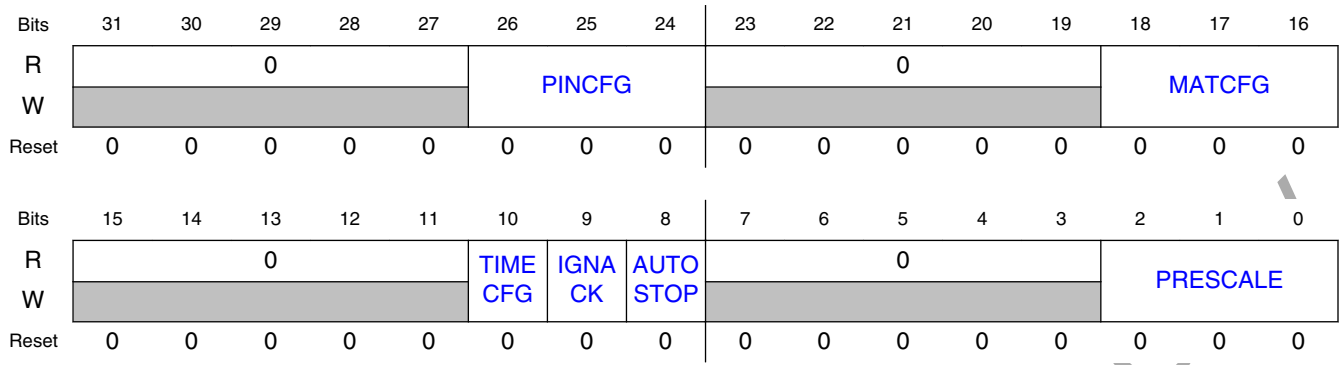
Register	Offset
MCFGR1	24h

43.3.1.9.2 Function

The MCFGR1 should only be written when the I2C Master is disabled.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.9.3 Diagram



43.3.1.9.4 Fields

Field	Function
31-27 —	Reserved
26-24 PINCFG	Pin Configuration Configures the pin mode. 000b - LPI2C configured for 2-pin open drain mode. 001b - LPI2C configured for 2-pin output only mode (ultra-fast mode). 010b - LPI2C configured for 2-pin push-pull mode. 011b - LPI2C configured for 4-pin push-pull mode. 100b - LPI2C configured for 2-pin open drain mode with separate LPI2C slave. 101b - LPI2C configured for 2-pin output only mode (ultra-fast mode) with separate LPI2C slave. 110b - LPI2C configured for 2-pin push-pull mode with separate LPI2C slave. 111b - LPI2C configured for 4-pin push-pull mode (inverted outputs).
23-19 —	Reserved
18-16 MATCFG	Match Configuration Configures the condition that will cause the DMF to set. 000b - Match disabled. 001b - Reserved. 010b - Match enabled (1st data word equals MATCH0 OR MATCH1). 011b - Match enabled (any data word equals MATCH0 OR MATCH1). 100b - Match enabled (1st data word equals MATCH0 AND 2nd data word equals MATCH1). 101b - Match enabled (any data word equals MATCH0 AND next data word equals MATCH1). 110b - Match enabled (1st data word AND MATCH1 equals MATCH0 AND MATCH1). 111b - Match enabled (any data word AND MATCH1 equals MATCH0 AND MATCH1).
15-11 —	Reserved
10 TIMECFG	Timeout Configuration 0b - Pin Low Timeout Flag will set if SCL is low for longer than the configured timeout. 1b - Pin Low Timeout Flag will set if either SCL or SDA is low for longer than the configured timeout.
9	IGNACK

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
IGNACK	When set, the received NACK field is ignored and assumed to be ACK. This bit is required to be set in Ultra-Fast Mode. 0b - LPI2C Master will receive ACK and NACK normally. 1b - LPI2C Master will treat a received NACK as if it was an ACK.
8 AUTOSTOP	Automatic STOP Generation When enabled, a STOP condition is generated whenever the LPI2C master is busy and the transmit FIFO is empty. The STOP condition can also be generated using a transmit FIFO command. 0b - No effect. 1b - STOP condition is automatically generated whenever the transmit FIFO is empty and LPI2C master is busy.
7-3 —	Reserved
2-0 PRESCALE	Prescaler Configures the clock prescaler used for all LPI2C master logic, except the digital glitch filters. 000b - Divide by 1. 001b - Divide by 2. 010b - Divide by 4. 011b - Divide by 8. 100b - Divide by 16. 101b - Divide by 32. 110b - Divide by 64. 111b - Divide by 128.

43.3.1.10 Master Configuration Register 2 (MCFGR2)

43.3.1.10.1 Address

Register	Offset
MCFGR2	28h

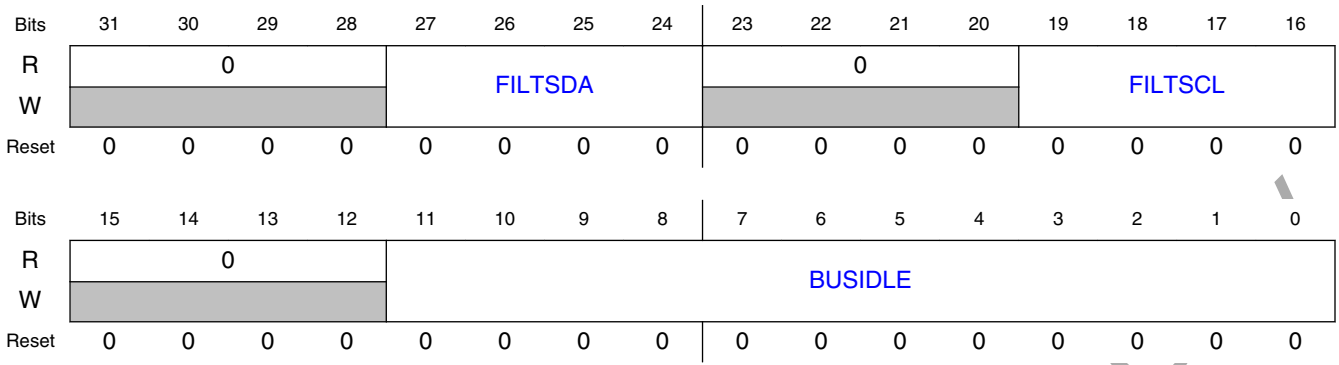
43.3.1.10.2 Function

The MCFGR2 should only be written when the I2C Master is disabled.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.10.3 Diagram



43.3.1.10.4 Fields

Field	Function
31-28 —	Reserved
27-24 FILTSDA	Glitch Filter SDA Configures the I2C master digital glitch filters for SDA input, a configuration of 0 will disable the glitch filter. Glitches equal to or less than FILTSDA cycles long will be filtered out and ignored. The latency through the glitch filter is equal to FILTSDA cycles and must be configured less than the minimum SCL low or high period. The glitch filter cycle count is not affected by the PRESCALE configuration and is automatically bypassed in High Speed mode.
23-20 —	Reserved
19-16 FILTSCL	Glitch Filter SCL Configures the I2C master digital glitch filters for SCL input, a configuration of 0 will disable the glitch filter. Glitches equal to or less than FILTSCL cycles long will be filtered out and ignored. The latency through the glitch filter is equal to FILTSCL cycles and must be configured less than the minimum SCL low or high period. The glitch filter cycle count is not affected by the PRESCALE configuration and is automatically bypassed in High Speed mode.
15-12 —	Reserved
11-0 BUSIDLE	Bus Idle Timeout Configures the bus idle timeout period in clock cycles. If both SCL and SDA are high for longer than BUSIDLE cycles, then the I2C bus is assumed to be idle and the master can generate a START condition. When set to zero, this feature is disabled.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.11 Master Configuration Register 3 (MCFGR3)

43.3.1.11.1 Address

Register	Offset
MCFGR3	2Ch

43.3.1.11.2 Function

The MCFGR3 should only be written when the I2C Master is disabled.

43.3.1.11.3 Diagram

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

43.3.1.11.4 Fields

Field	Function
31-20 —	Reserved
19-8 PINLOW	Pin Low Timeout Configures the pin low timeout flag in clock cycles. If SCL and/or SDA is low for longer than (PINLOW * 256) cycles then PLTF is set. When set to zero, this feature is disabled.
7-0 —	Reserved

43.3.1.12 Master Data Match Register (MDMR)

Freescale Confidential Proprietary Use Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.12.1 Address

Register	Offset
MDMR	40h

43.3.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								MATCH1							
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								MATCH0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.12.3 Fields

Field	Function
31-24 —	Reserved
23-16 MATCH1	Match 1 Value Compared against the received data when receive data match is enabled.
15-8 —	Reserved
7-0 MATCH0	Match 0 Value Compared against the received data when receive data match is enabled.

43.3.1.13 Master Clock Configuration Register 0 (MCCR0)

43.3.1.13.1 Address

Register	Offset
MCCR0	48h

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.13.2 Function

The MCCR0 cannot be changed when the I2C master is enabled and is used for standard, fast, fast-mode plus and ultra-fast transfers.

43.3.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DATAVD						0		SETHOLD					
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		CLKHI						0		CLKLO					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.13.4 Fields

Field	Function
31-30 —	Reserved
29-24 DATAVD	Data Valid Delay Minimum number of cycles (minus one) that is used as the data hold time for SDA. Must be configured less than the minimum SCL low period.
23-22 —	Reserved
21-16 SETHOLD	Setup Hold Delay Minimum number of cycles (minus one) that is used by the master as the setup and hold time for a (repeated) START condition and setup time for a STOP condition. The setup time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this is equal to (2 + FILTSCL) / 2^PRESCALE cycles.
15-14 —	Reserved
13-8 CLKHI	Clock High Period Minimum number of cycles (minus one) that the SCL clock is driven high by the master. The SCL high time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this is equal to (2 + FILTSCL) / 2^PRESCALE cycles.
7-6 —	Reserved
5-0 CLKLO	Clock Low Period Minimum number of cycles (minus one) that the SCL clock is driven low by the master. This value is also used for the minimum bus free time between a STOP and a START condition.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.14 Master Clock Configuration Register 1 (MCCR1)

43.3.1.14.1 Address

Register	Offset
MCCR1	50h

43.3.1.14.2 Function

The MCCR1 cannot be changed when the I2C master is enabled and is used for high speed mode transfers. The separate clock configuration for high speed mode allows arbitration to take place in Fast mode (with timing configured by MCCR0), before switching to high speed mode (with timing configured by MCCR1).

43.3.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DATAVD						0		SETHOLD					
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		CLKHI						0		CLKLO					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.14.4 Fields

Field	Function
31-30 —	Reserved
29-24 DATAVD	Data Valid Delay Minimum number of cycles (minus one) that is used as the data hold time for SDA. Must be configured less than the minimum SCL low period.
23-22 —	Reserved
21-16	Setup Hold Delay

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
SETHOLD	Minimum number of cycles (minus one) that is used by the master as the setup and hold time for a (repeated) START condition and setup time for a STOP condition. The setup time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this is equal to (2 + FILTSCL) / 2^PRESCALE cycles.
15-14 —	Reserved
13-8 CLKHI	Clock High Period Minimum number of cycles (minus one) that the SCL clock is driven high by the master. The SCL high time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this is equal to (2 + FILTSCL) / 2^PRESCALE cycles.
7-6 —	Reserved
5-0 CLKLO	Clock Low Period Minimum number of cycles (minus one) that the SCL clock is driven low by the master. This value is also used for the minimum bus free time between a STOP and a START condition.

43.3.1.15 Master FIFO Control Register (MFCR)

43.3.1.15.1 Address

Register	Offset
MFCR	58h

43.3.1.15.2 Diagram

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.15.3 Fields

Field	Function
31-24 —	Reserved
23-16 RXWATER	Receive FIFO Watermark The Receive Data Flag is set whenever the number of words in the receive FIFO is greater than RXWATER. Writing a value equal or greater than the FIFO size will be truncated.
15-8 —	Reserved
7-0 TXWATER	Transmit FIFO Watermark The Transmit Data Flag is set whenever the number of words in the transmit FIFO is equal or less than TXWATER. Writing a value equal or greater than the FIFO size will be truncated.

43.3.1.16 Master FIFO Status Register (MFSR)

43.3.1.16.1 Address

Register	Offset
MFSR	5Ch

43.3.1.16.2 Diagram

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

43.3.1.16.3 Fields

Field	Function
31-24	Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
—	
23-16 RXCOUNT	Receive FIFO Count Returns the number of words in the receive FIFO.
15-8 —	Reserved
7-0 TXCOUNT	Transmit FIFO Count Returns the number of words in the transmit FIFO.

43.3.1.17 Master Transmit Data Register (MTDR)

43.3.1.17.1 Address

Register	Offset
MTDR	60h

43.3.1.17.2 Function

An 8-bit write to the CMD field will store the data in the Command FIFO, but does not increment the FIFO write pointer. An 8-bit write to the DATA field will zero extend the CMD field unless the CMD field has been written separately since the last FIFO write, it also increments the FIFO write pointer. A 16-bit or 32-bit will write both the CMD and DATA fields and increment the FIFO.

43.3.1.17.3 Diagram

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

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.17.4 Fields

Field	Function
31-11 Reserved	Reserved
10-8 CMD	Command Data 000b - Transmit DATA[7:0]. 001b - Receive (DATA[7:0] + 1) bytes. 010b - Generate STOP condition. 011b - Receive and discard (DATA[7:0] + 1) bytes. 100b - Generate (repeated) START and transmit address in DATA[7:0]. 101b - Generate (repeated) START and transmit address in DATA[7:0]. This transfer expects a NACK to be returned. 110b - Generate (repeated) START and transmit address in DATA[7:0] using high speed mode. 111b - Generate (repeated) START and transmit address in DATA[7:0] using high speed mode. This transfer expects a NACK to be returned.
7-0 DATA	Transmit Data Performing an 8-bit write to DATA will zero extend the CMD field.

43.3.1.18 Master Receive Data Register (MRDR)

43.3.1.18.1 Address

Register	Offset
MRDR	70h

43.3.1.18.2 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	RXEMPTY	0						DATA							
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.18.3 Fields

Field	Function
31-15 Reserved	Reserved
14 RXEMPTY	RX Empty 0b - Receive FIFO is not empty. 1b - Receive FIFO is empty.
13-8 —	Reserved
7-0 DATA	Receive Data Reading this register returns the data received by the I2C master that has not been discarded. Receive data can be discarded due to the CMD field or the master can be configured to discard non-matching data.

43.3.1.19 Slave Control Register (SCR)

43.3.1.19.1 Address

Register	Offset
SCR	110h

43.3.1.19.2 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						0	0	0		FILTDZ	FILTEN	0		RST	SEN
W							RRF	RTF								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.19.3 Fields

Field	Function
31-10	Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
Reserved	
9 RRF	Reset Receive FIFO 0b - No effect. 1b - Receive Data Register is now empty.
8 RTF	Reset Transmit FIFO 0b - No effect. 1b - Transmit Data Register is now empty.
7-6 —	Reserved
5 FILTDZ	Filter Doze Enable 0b - Filter remains enabled in Doze mode. 1b - Filter is disabled in Doze mode.
4 FILTEN	Filter Enable 0b - Disable digital filter and output delay counter for slave mode. 1b - Enable digital filter and output delay counter for slave mode.
3-2 —	Reserved
1 RST	Software Reset 0b - Slave logic is not reset. 1b - Slave logic is reset.
0 SEN	Slave Enable 0b - Slave mode is disabled. 1b - Slave mode is enabled.

43.3.1.20 Slave Status Register (SSR)

43.3.1.20.1 Address

Register	Offset
SSR	114h

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BBF	SBF	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	SARF	GCF	AM1F	AM0F	FEF	BEF	SDF	RSF	0				TAF	AVF	RDF	TDF
W					w1c	w1c	w1c	w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.20.3 Fields

Field	Function
31-26 —	Reserved
25 BBF	Bus Busy Flag 0b - I2C Bus is idle. 1b - I2C Bus is busy.
24 SBF	Slave Busy Flag 0b - I2C Slave is idle. 1b - I2C Slave is busy.
23-16 —	Reserved
15 SARF	SMBus Alert Response Flag This flag is cleared by reading the Address Status Register. This flag cannot generate an asynchronous wakeup. 0b - SMBus Alert Response disabled or not detected. 1b - SMBus Alert Response enabled and detected.
14 GCF	General Call Flag This flag is cleared by reading the Address Status Register. This flag cannot generate an asynchronous wakeup. 0b - Slave has not detected the General Call Address or General Call Address disabled. 1b - Slave has detected the General Call Address.
13 AM1F	Address Match 1 Flag Indicates that the received address has matched the ADDR1 field or ADDR0 to ADDR1 range as configured by ADDRCFG. This flag is cleared by reading the Address Status Register. This flag cannot generate an asynchronous wakeup. 0b - Have not received ADDR1 or ADDR0/ADDR1 range matching address. 1b - Have received ADDR1 or ADDR0/ADDR1 range matching address.
12 AM0F	Address Match 0 Flag Indicates that the received address has matched the ADDR0 field as configured by ADDRCFG. This flag is cleared by reading the Address Status Register. This flag cannot generate an asynchronous wakeup. 0b - Have not received ADDR0 matching address. 1b - Have received ADDR0 matching address.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
11 FEF	FIFO Error Flag FIFO error flag can only set when clock stretching is disabled. 0b - FIFO underflow or overflow not detected. 1b - FIFO underflow or overflow detected.
10 BEF	Bit Error Flag This flag will set if the LPI2C slave transmits a logic one and detects a logic zero on the I2C bus. The slave will ignore the rest of the transfer until the next (repeated) START condition. 0b - Slave has not detected a bit error. 1b - Slave has detected a bit error.
9 SDF	STOP Detect Flag This flag will set when the LPI2C slave detects a STOP condition, provided the LPI2C slave matched the last address byte. 0b - Slave has not detected a STOP condition. 1b - Slave has detected a STOP condition.
8 RSF	Repeated Start Flag This flag will set when the LPI2C slave detects a repeated START condition, provided the LPI2C slave matched the last address byte. It does not set when the slave first detects a START condition. 0b - Slave has not detected a Repeated START condition. 1b - Slave has detected a Repeated START condition.
7-4 —	Reserved
3 TAF	Transmit ACK Flag This flag is cleared by writing the transmit ACK register. 0b - Transmit ACK/NACK is not required. 1b - Transmit ACK/NACK is required.
2 AVF	Address Valid Flag This flag is cleared by reading the address status register. When RXCFG is set, this flag is also cleared by reading the receive data register. 0b - Address Status Register is not valid. 1b - Address Status Register is valid.
1 RDF	Receive Data Flag This flag is cleared by reading the receive data register. When RXCFG is set, this flag is not cleared when reading the receive data register and AVF is set. 0b - Receive Data is not ready. 1b - Receive data is ready.
0 TDF	Transmit Data Flag This flag is cleared by writing the transmit data register. When TXCFG is clear, it is also cleared if a NACK or Repeated START or STOP condition is detected. 0b - Transmit data not requested. 1b - Transmit data is requested.

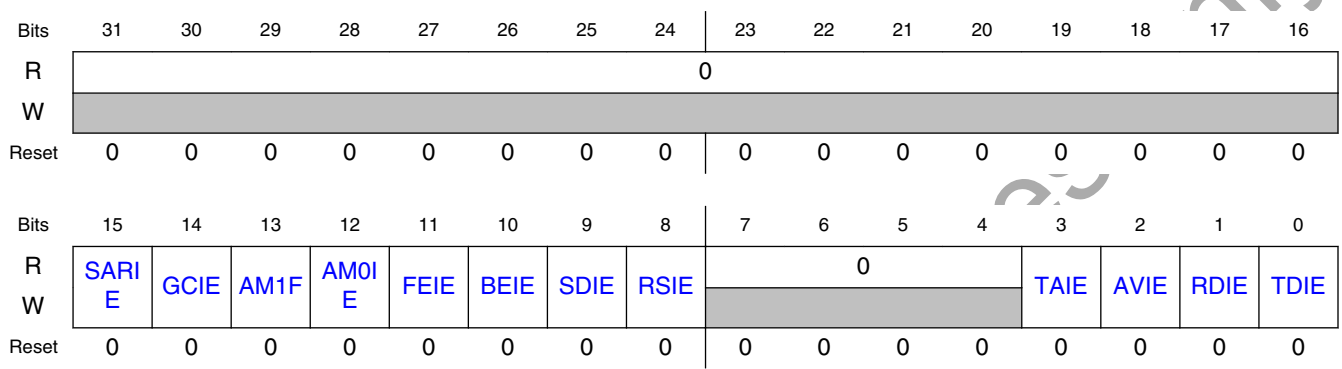
43.3.1.21 Slave Interrupt Enable Register (SIER)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.21.1 Address

Register	Offset
SIER	118h

43.3.1.21.2 Diagram



43.3.1.21.3 Fields

Field	Function
31-16 —	Reserved
15 SARIE	SMBus Alert Response Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
14 GCIE	General Call Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
13 AM1F	Address Match 1 Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
12 AM0IE	Address Match 0 Interrupt Enable 0b - Interrupt enabled. 1b - Interrupt disabled.
11 FEIE	FIFO Error Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
10 BEIE	Bit Error Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
9 SDIE	STOP Detect Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

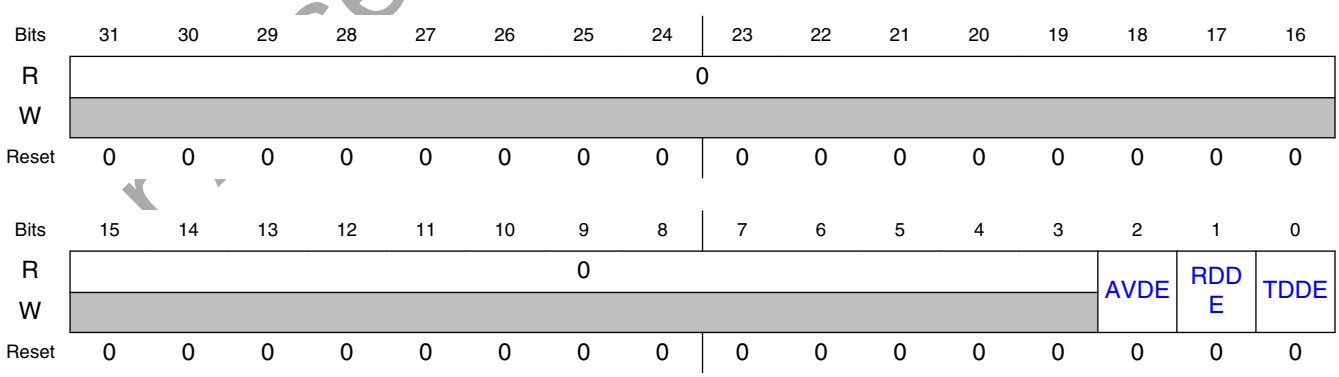
Field	Function
8 RSIE	Repeated Start Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
7-4 —	Reserved
3 TAIE	Transmit ACK Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
2 AVIE	Address Valid Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
1 RDIE	Receive Data Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled.
0 TDIE	Transmit Data Interrupt Enable 0b - Interrupt disabled. 1b - Interrupt enabled

43.3.1.22 Slave DMA Enable Register (SDER)

43.3.1.22.1 Address

Register	Offset
SDER	11Ch

43.3.1.22.2 Diagram



Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.22.3 Fields

Field	Function
31-3 Reserved	Reserved
2 AVDE	Address Valid DMA Enable The Address Valid DMA request is shared with the Receive Data DMA request. If both are enabled, then set RXCFG to allow the DMA to read the address from the Receive Data Register. 0b - DMA request disabled. 1b - DMA request enabled.
1 RDDE	Receive Data DMA Enable 0b - DMA request disabled. 1b - DMA request enabled.
0 TDDE	Transmit Data DMA Enable 0b - DMA request disabled. 1b - DMA request enabled

43.3.1.23 Slave Configuration Register 1 (SCFGR1)

43.3.1.23.1 Address

Register	Offset
SCFGR1	124h

43.3.1.23.2 Function

The SCFGR1 should only be written when the I2C Slave is disabled.

43.3.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													ADDRCFG		
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		HSMEN	IGNACK	RXCFG	TXCFG	SAEN	GCEN	0				ACKSTALL	TXDSTALL	RXSTALL	ADRSTALL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.23.4 Fields

Field	Function
31-19 —	Reserved
18-16 ADDRCFG	Address Configuration Configures the condition that will cause an address to match. 000b - Address match 0 (7-bit). 001b - Address match 0 (10-bit). 010b - Address match 0 (7-bit) or Address match 1 (7-bit). 011b - Address match 0 (10-bit) or Address match 1 (10-bit). 100b - Address match 0 (7-bit) or Address match 1 (10-bit). 101b - Address match 0 (10-bit) or Address match 1 (7-bit). 110b - From Address match 0 (7-bit) to Address match 1 (7-bit). 111b - From Address match 0 (10-bit) to Address match 1 (10-bit).
15-14 —	Reserved
13 HSMEN	High Speed Mode Enable Enables detection of the High-speed Mode master code of slave address 0000_1XX, but does not cause an address match on this code. When set and any Hs-mode master code is detected, the FILTEN and ACKSTALL bits are ignored until the next STOP condition is detected. 0b - Disables detection of Hs-mode master code. 1b - Enables detection of Hs-mode master code.
12 IGNACK	Ignore NACK When set, the LPI2C slave will continue transfers after a NACK is detected. This bit is required to be set in Ultra-Fast Mode. 0b - Slave will end transfer when NACK detected. 1b - Slave will not end transfer when NACK detected.
11 RXCFG	Receive Data Configuration 0b - Reading the receive data register will return receive data and clear the receive data flag. 1b - Reading the receive data register when the address valid flag is set will return the address status register and clear the address valid flag. Reading the receive data register when the address valid flag is clear will return receive data and clear the receive data flag.
10 TXCFG	Transmit Flag Configuration The transmit data flag will always assert before a NACK is detected at the end of a slave-transmit transfer. This can cause an extra word to be written to the transmit data FIFO. When TXCFG=0, the transmit data register is automatically emptied when a slave-transmit transfer is detected. This cause the transmit data flag to assert whenever a slave-transmit transfer is detected and negate at the end of the slave-transmit transfer. When TXCFG=1, the transmit data flag will assert whenver the transit data register is empty and negate when the transmit data register is full. This allows the transmit data register to be filled before a slave-transmit transfer is detected, but can cause the transmit data register to be written before a NACK is detected on the last byte of a slave transmit transfer. 0b - Transmit Data Flag will only assert during a slave-transmit transfer when the transmit data register is empty. 1b - Transmit Data Flag will assert whenever the transmit data register is empty.
9 SAEN	SMBus Alert Enable 0b - Disables match on SMBus Alert. 1b - Enables match on SMBus Alert.
8	General Call Enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
GCEN	0b - General Call address is disabled. 1b - General call address is enabled.
7-4 —	Reserved
3 ACKSTALL	ACK SCL Stall Enables SCL clock stretching during slave-transmit address byte(s) and slave-receiver address and data byte(s) to allow software to write the Transmit ACK Register before the ACK or NACK is transmitted. Clock stretching occurs when transmitting the 9th bit and is therefore not compatible with high speed mode. When ACKSTALL is enabled, there is no need to set either RXSTALL or ADRSTALL 0b - Clock stretching disabled. 1b - Clock stretching enabled.
2 TXDSTALL	TX Data SCL Stall Enables SCL clock stretching when the transmit data flag is set during a slave-transmit transfer. Clock stretching occurs following the 9th bit and is therefore compatible with high speed mode. 0b - Clock stretching disabled. 1b - Clock stretching enabled.
1 RXSTALL	RX SCL Stall Enables SCL clock stretching when receive data flag is set during a slave-receive transfer. Clock stretching occurs following the 9th bit and is therefore compatible with high speed mode. 0b - Clock stretching disabled. 1b - Clock stretching enabled.
0 ADRSTALL	Address SCL Stall Enables SCL clock stretching when the address valid flag is asserted. Clock stretching only occurs following the 9th bit and is therefore compatible with high speed mode. 0b - Clock stretching disabled. 1b - Clock stretching enabled.

43.3.1.24 Slave Configuration Register 2 (SCFGR2)

43.3.1.24.1 Address

Register	Offset
SCFGR2	128h

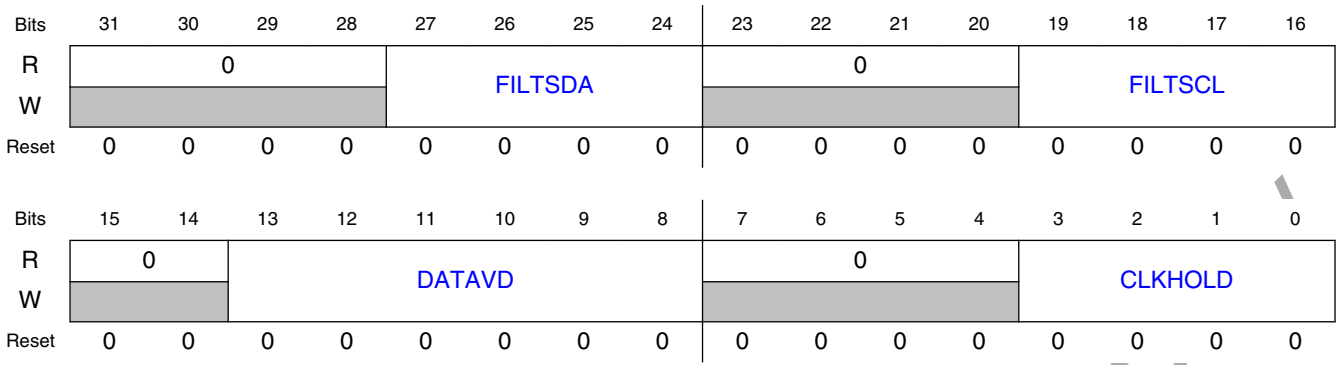
43.3.1.24.2 Function

The SCFGR2 should only be written when the I2C Slave is disabled.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.24.3 Diagram



43.3.1.24.4 Fields

Field	Function
31-28 —	Reserved
27-24 FILTSDA	Glitch Filter SDA Configures the I2C slave digital glitch filters for SDA input, a configuration of 0 will disable the glitch filter. Glitches equal to or less than FILTSDA cycles long will be filtered out and ignored. The latency through the glitch filter is equal to FILTSDA+3 cycles and must be configured less than the minimum SCL low or high period. The glitch filter cycle count is not affected by the PRESCALE configuration, and is disabled in high speed mode.
23-20 —	Reserved
19-16 FILTSCL	Glitch Filter SCL Configures the I2C slave digital glitch filters for SCL input, a configuration of 0 will disable the glitch filter. Glitches equal to or less than FILTSCL cycles long will be filtered out and ignored. The latency through the glitch filter is equal to FILTSCL+3 cycles and must be configured less than the minimum SCL low or high period. The glitch filter cycle count is not affected by the PRESCALE configuration, and is disabled in high speed mode.
15-14 —	Reserved
13-8 DATAVD	Data Valid Delay Configures the SDA data valid delay time for the I2C slave equal to FILTSCL+DATAVD+3 cycles. This data valid delay must be configured to less than the minimum SCL low period. The I2C slave data valid delay time is not affected by the PRESCALE configuration, and is disabled in high speed mode.
7-4 —	Reserved
3-0	Clock Hold Time

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
CLKHOLD	Configures the minimum clock hold time for the I2C slave, when clock stretching is enabled. The minimum hold time is equal to CLKHOLD+3 cycles. The I2C slave clock hold time is not affected by the PRESCALE configuration, and is disabled in high speed mode.

43.3.1.25 Slave Address Match Register (SAMR)

43.3.1.25.1 Address

Register	Offset
SAMR	140h

43.3.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					ADDR1										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					ADDR0										0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.25.3 Fields

Field	Function
31-27 —	Reserved
26-17 ADDR1	Address 1 Value Compared against the received address to detect the Slave Address. In 10-bit mode, the first address byte is compared to { 11110, ADDR1[10:9] } and the second address byte is compared to ADDR1[8:1]. In 7-bit mode, the address is compared to ADDR1[7:1].
16-11 Reserved	Reserved
10-1	Address 0 Value

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
ADDR0	Compared against the received address to detect the Slave Address. In 10-bit mode, the first address byte is compared to { 11110, ADDR0[10:9] } and the second address byte is compared to ADDR0[8:1]. In 7-bit mode, the address is compared to ADDR0[7:1].
0 —	Reserved

43.3.1.26 Slave Address Status Register (SASR)

43.3.1.26.1 Address

Register	Offset
SASR	150h

43.3.1.26.2 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	ANV	0			RADDR										
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.26.3 Fields

Field	Function
31-15 Reserved	Reserved
14 ANV	Address Not Valid 0b - RADDR is valid. 1b - RADDR is not valid.
13-11 —	Reserved
10-0	Received Address

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
RADDR	RADDR updates whenever the AMF is set and the AMF is cleared by reading this register. In 7-bit mode, the address byte is store in RADDR[7:0]. In 10-bit mode, the first address byte is { 11110, RADDR[10:9], RADDR[0] } and the second address byte is RADDR[8:1]. The R/W bit is therefore always stored in RADDR[0].

43.3.1.27 Slave Transmit ACK Register (STAR)

43.3.1.27.1 Address

Register	Offset
STAR	154h

43.3.1.27.2 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															TXNACK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.3.1.27.3 Fields

Field	Function
31-1 Reserved	Reserved
0 TXNACK	Transmit NACK When NACKSTALL is set, must be written once for each matching address byte and each received word. Can also be written when LPI2C Slave is disabled or idle to configure the default ACK/NACK. 0b - Transmit ACK for received word. 1b - Transmit NACK for received word.

Freescale Confidential Proprietary Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.28 Slave Transmit Data Register (STDR)

43.3.1.28.1 Address

Register	Offset
STDR	160h

43.3.1.28.2 Diagram

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

43.3.1.28.3 Fields

Field	Function
31-8 Reserved	Reserved
7-0 DATA	Transmit Data Writing this register will store I2C slave transmit data in the transmit register.

43.3.1.29 Slave Receive Data Register (SRDR)

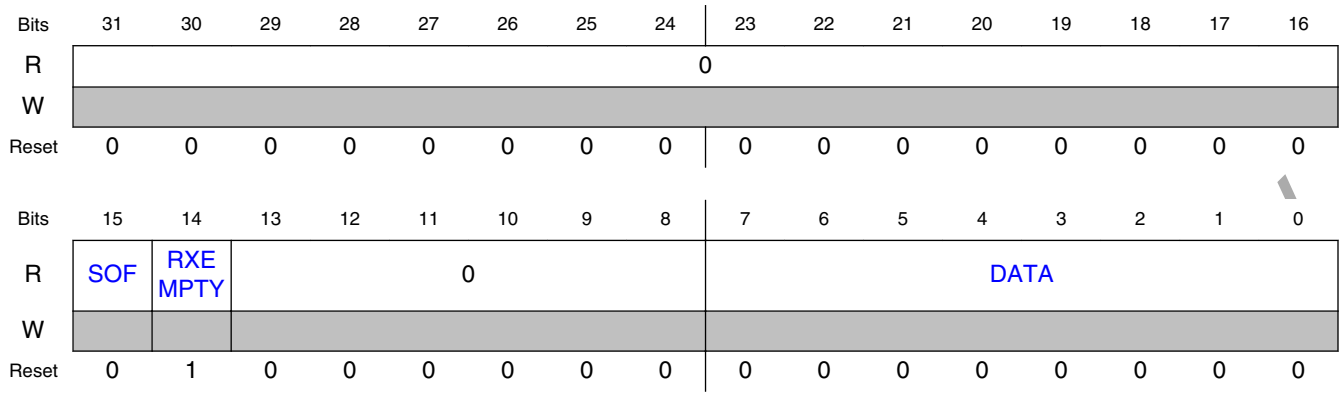
43.3.1.29.1 Address

Register	Offset
SRDR	170h

For Assessment Purpose Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.3.1.29.2 Diagram



43.3.1.29.3 Fields

Field	Function
31-16 —	Reserved
15 SOF	Start Of Frame 0b - Indicates this is not the first data word since a (repeated) START or STOP condition. 1b - Indicates this is the first data word since a (repeated) START or STOP condition.
14 RXEMPTY	RX Empty 0b - The Receive Data Register is not empty. 1b - The Receive Data Register is empty.
13-8 —	Reserved
7-0 DATA	Receive Data Reading this register returns the data received by the I2C slave.

43.4 Functional description

43.4.1 Clocking and Resets

For Asses

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.4.1.1 Functional clock

The LPI2C functional clock is asynchronous to the bus clock and can remain enabled in low power modes to support I2C bus transfers by the LPI2C master. It is also used by the LPI2C slave to support digital filter and data hold time configurations. The LPI2C master divides the functional clock by a prescaler and the resulting frequency must be at least eight times faster than the I2C bus bandwidth.

43.4.1.2 External clock

The LPI2C slave logic is clocked directly from the external pins SCL and SDA (or SCLS and SDAS if master and slave are implemented on separate pins). This allows the LPI2C slave to remain operational, even when the LPI2C functional clock is disabled. Note that the LPI2C slave digital filter must be disabled if the LPI2C functional clock is disabled and this can effect compliance with some of the timing parameters of the I2C specification, such as the data hold time.

43.4.1.3 Bus clock

The bus clock is only used for bus accesses to the control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the LPI2C master and slave registers.

43.4.1.4 Chip reset

The logic and registers for the LPI2C master and slave are reset to their default state on a chip reset.

43.4.1.5 Software reset

The LPI2C master implements a software reset bit in its Control Register. The MCR[RST] will reset all master logic and registers to their default state, except for the MCR itself.

The LPI2C slave implements a software reset bit in its Control Register. The SCR[RST] will reset all slave logic and registers to their default state, except for the SCR itself.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.4.1.6 FIFO reset

The LPI2C master implements write-only control bits that resets the transmit FIFO (MCR[RTF] and receive FIFO (MCR[RRF]). A FIFO is empty after being reset.

The LPI2C slave implements write-only control bits that resets the transmit data register (SCR[RTF] and receive data register (SCR[RRF]). A data register is empty after being reset.

43.4.2 Master Mode

The LPI2C master logic operates independently from the slave logic to perform all master mode transfers on the I2C bus.

43.4.2.1 Transmit and Command FIFO

The transmit FIFO stores command data to initiate the various I2C operations. The following operations can be initiated through commands in the transmit FIFO:

- START or Repeated START condition with address byte and expecting ACK or NACK.
- Transmit data (this is the default for zero extended byte writes to the transmit FIFO).
- Receive 1-256 bytes of data (can also be configured to discard receive data and not store in receive FIFO).
- STOP condition (can also be configured to send STOP condition when transmit FIFO is empty).

Multiple transmit and receive commands can be inserted between the START condition and STOP conditon, transmit and receive commands must not be interleaved in order to comply with the I2C specification. The receive data command and the receive data and discard command can be interleaved to ensure only the desired received data is stored in the receive FIFO (or compared with the data match logic).

The LPI2C master supports 10-bit addressing through a (repeated) START condition, followed by a transmit byte with the second address byte, followed by any number of data bytes with the master-transmit data.

A START or Repeated START condition that is expecting a NACK (for example, hs-mode master code) must be followed by a STOP or (repeated) START condition.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.4.2.2 Master Operation

Whenever the LPI2C is enabled, it monitors the I2C bus to detect when the I2C bus is idle (MSR[BBF]). The I2C bus is no longer considered idle if either SCL or SDA are low and becomes idle if a STOP condition is detected or if a bus idle timeout is detected (as configured by MCFGR2[BUSIDLE]). Once the I2C bus is idle, the transmit FIFO is not empty, and the host request is either asserted or disabled, then the LPI2C master will initiate a transfer on the I2C bus. This involves the following steps:

- Wait the bus idle time equal to (MCCR0[CLKLO] + 1) multiplied by the prescaler.
- Transmit a START condition and address byte using the timing configuration in MCCR0, if a high speed mode transfer is configured then timing configuration from MCCR1 is used instead.
- Perform master-transmit or master-receive transfers, as configured by the transmit FIFO.
- Transmit a Repeated START or STOP condition as configured by the transmit FIFO and/or MCFGR1[AUTOSTOP]. A repeated START can change which timing configuration register is used.

When the LPI2C master is disabled (either due to MCR[MEN] being clear or automatically due to mode entry), the LPI2C will continue to empty the transmit FIFO until a STOP condition is transmitted. However, it will no longer stall the I2C bus waiting for the transmit or receive FIFO and once the transmit FIFO is empty it will generate a STOP condition automatically.

The LPI2C master can stall the I2C bus under certain conditions, this will result in SCL pulled low continuously on the first bit of a byte until the condition is removed:

- LPI2C master is enabled and busy, transmit FIFO is empty, and MCFGR1[AUTOSTOP] is clear.
- LPI2C master is enabled and receiving data, receive data is not being discarded (due to command or receive data match), and receive FIFO is full.

43.4.2.3 Receive FIFO and Data Match

The receive FIFO is used to store receive data during master-receiver transfers. Receive data can also be configured to discard receive data instead of storing in the receive FIFO, this is configured by the command word in the transmit FIFO.

Receive data supports a receive data match function that can match received data against one of two bytes or against a masked data byte. The data match function can also be configured to compare only the first one or two received data words since the last

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

(repeated) START condition. Receive data that is already discarded due to the command word cannot cause the data match to set and will delay the match on first received data word until after the discarded data is received. The receiver match function can also be configured to discard all receive data until a data match is detected, using the MCFGR0[RDMO] control bit. When clearing the MCFGR0[RDMO] control bit following a data match, clear MCFGR0[RDMO] before clearing MSR[DMF] to allow all subsequent data to be received.

43.4.2.4 Timing Parameters

The following timing parameters can be configured by the LPI2C master. Parameters are configured separately for high speed mode (MCCR1) and other modes (MCCR0). This allows the high speed mode master code to be sent using the regular timing parameters and then switch to the high speed mode timing (following a repeated START) until the next STOP condition.

The LPI2C master timing parameters in LPI2C functional clock cycles are configured as follows. They must be configured to meet the I2C timing specification for the required mode.

Table 43-3. Timing Parameters

I2C Specification Timing Parameter	I2C Specification Timing Symbol	LPI2C Timing Parameter (LPI2C functional clock cycles)
hold time (repeated) START condition	tHD:STA	(SETHOLD +1) x (2 ^ PRESCALE)
LOW period of the SCL clock	tLOW	(CLKLO + 1) x (2 ^ PRESCALE)
HIGH period of the SCL clock	tHIGH	(CLKHI + 1 + SCL_LATENCY) x (2 ^ PRESCALE)
setup time for a repeated START condition or STOP condition	tSU:STA, tSU:STO	(SETHOLD + 1 + SCL_LATENCY) x (2 ^ PRESCALE)
data hold time	tHD:DAT	(DATAVD + 1) x (2 ^ PRESCALE)
data setup time	tSU:DAT	(SDA_LATENCY + 1) x (2 ^ PRESCALE)
bus free time between a STOP and START condition	tBUF	(CLKLO + 1 + SDA_LATENCY) x (2 ^ PRESCALE)
data valid time, data valid acknowledge time	tVD:DAT, tVD:ACK	(DATAVD + 1) x (2 ^ PRESCALE) + tr:DAT

- SCL_LATENCY is the external SCL rise time plus the time it takes to propagate through the SCL synchronizer and glitch filter.
- SDA_LATENCY is the external SDA rise time plus the time it takes to propagate through the SDA synchronizer and glitch filter.

For Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The latency depends on a number of factors including the bus loading and external pull-up resistor sizing. The minimum delay equals two plus the pin input digital filter setting (which are configured separately for SCL and SDA), divided by the prescaler (since the pin input digital filters are not affected by the prescaler setting).

The following timing restrictions must be enforced to avoid unexpected START or STOP conditions on the I2C bus or unexpected START or STOP conditions detected by the LPI2C master. They can be summarized as SDA cannot change when SCL is high outside of a transmitted (repeated) START or STOP condition.

Table 43-4. LPI2C Timing Parameter Restrictions

Timing Parameter	Minimum	Maximum	Comment
CLKLO	0x03	-	CLKLO must also be greater than delay through the SCL filter.
CLKHI	0x01	-	
SETHOLD	0x02	-	
DATAVD	0x01	CLKLO - [(FILTSDA+2) / (2 ^ PRESCALE)]	DATAVD must be less than CLKLO minus delay through the SDA filter.
FILTSCL	0x00	[CLKLO × (2 ^ PRESCALE)] - 3	
FILTSDA	FILTSCL	[CLKLO × (2 ^ PRESCALE)] - 3	Does not apply if compensating for board level skew between SCL and SDA.
BUSIDLE	(CLKLO+SETHOLD+2) × 2	-	Must also be greater than CLKHI+1.

The timing parameters must be configured to meet the requirements of the I2C specification, this will depend on the mode being supported, the frequency of the LPI2C functional clock. Some example timing configurations are provided below.

Table 43-5. LPI2C Example Timing Configurations

I2C Mode	Clock Frequency	Baud Rate	PRESCALE	FILTSCL/ FILTSDA	SETHOLD	CLKLO	CLKHI	DATAVD
Fast	8 MHz	400 kbps	0x0	0x0/0x0	0x04	0x0B	0x05	0x02
Fast+	8 MHz	1 Mbps	0x0	0x0/0x0	0x02	0x03	0x01	0x01
Fast	48 MHz	400 kbps	0x2	0x1/0x1	0x07	0x11	0x0B	0x03
Fast+	48 MHz	1 Mbps	0x2	0x1/0x1	0x03	0x06	0x04	0x04
Fast+	48 MHz	1 Mbps	0x0	0x1/0x1	0x1D	0x3E	0x35	0x0F
HS-mode	48 MHz	3.2 Mbps	0x0	0x0/0x0	0x07	0x08	0x03	0x01
Fast	60 MHz	400 kbps	0x1	0x2/0x2	0x11	0x28	0x1F	0x08
Fast+	60 MHz	1 Mbps	0x1	0x2/0x2	0x07	0x0F	0x0B	0x01
HS-mode	60 MHz	3.33 Mbps	0x1	0x0/0x0	0x04	0x04	0x02	0x01

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.4.2.5 Error Conditions

The LPI2C master will monitor for errors while it is active, the following conditions will generate an error flag and block a new START condition from being sent until the flag is cleared by software:

- START or STOP condition detected and not generated by LPI2C master (sets MSR[ALF]).
- Transmitting data on SDA and different value being received (sets MSR[ALF]).
- NACK detected when transmitting data, provided MCFGR1[IGNACK] is clear (sets MSR[NDF]).
- NACK detected and expecting ACK for address byte, provided MCFGR1[IGNACK] is clear (sets MSR[NDF]).
- ACK detected and expecting NACK for address byte, provided MCFGR1[IGNACK] is clear (sets MSR[NDF]).
- Transmit FIFO requesting to transmit or receive data without a START condition (sets MSR[FEF]).
- SCL (or SDA if MCFGR1[TIMECFG] is set) is low for (MCFGR2[TIMELOW] * 256) prescaler cycles without a pin transition (sets MSR[PLTF]).

Software must respond to the MSR[PTLF] flag to terminate the existing command either cleanly (by clearing MCR[MEN]) or abruptly (by setting MCR[SWRST]).

The MCFGR2[BUSIDLE] field can be used to force the I2C bus to be considered idle when SCL and SDA remain high for (BUSIDLE+1) prescaler cycles. The I2C bus is normally considered idle when the LPI2C master is first enabled, but when BUSIDLE is configured greater than zero then SCL and/or SDA must be high for (BUSIDLE+1) prescaler cycles before the I2C bus is first considered idle.

43.4.2.6 Pin Configuration

The LPI2C master defaults to open-drain configuration of the SDA and SCL pins. Support for true open drain is device specific and requires the pins where LPI2C pins are muxed to support true open drain. Support for high speed mode is also device specific and requires the SCL pin to support the current source pull-up required in the I2C specification.

The LPI2C master also supports the output only push-pull function required for I2C ultra-fast mode using the SDA and SCL pins. Support for ultra-fast mode also requires the IGNACK bit to be set.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

A push-pull 2 wire configuration is also available to the LPI2C master that may support a partial high speed mode provided the LPI2C is the only master and all I2C pins on the bus are at the same voltage. This will configure the SCL pin as push-pull for every clock except the 9th clock pulse to allow high speed mode compatible slaves to perform clock stretching. In this mode, the SDA pin is tristated for master-receive data bits and master-transmit ACK/NACK bits.

The push-pull 4 wire configuration separates the SCL input and output and the SDA input and output onto separate pins, with SCL/SDA used as the input pins and SCLS/SDAS used as the output pins with configurable polarity. This simplifies the external connections when connecting the I2C bus to external level shifters. The LPI2C master logic and LPI2C slave logic are not able to connect to separate I2C buses when using this configuration.

43.4.3 Slave Mode

The LPI2C slave logic operates independently from the master logic to perform all slave mode transfers on the I2C bus.

43.4.3.1 Address Match

The LPI2C slave can be configured to match one of two addresses using either 7-bit or 10-bit addressing modes for each address, or to match a range of addresses in either 7-bit or 10-bit addressing modes. Separately, it can be configured to match the General Call Address or the SMBus Alert Address and generate appropriate flags. The LPI2C slave can also be configured to detect the high speed mode master code and to disable the digital filters and output valid delay time until the next STOP condition is detected.

Once a valid address is matched, the LPI2C slave will automatically perform slave-transmit or slave-receive transfers until a NACK is detected (unless IGNACK is set), a bit error is detected (the LPI2C slave is driving SDA, but a different value is sampled), or a (repeated) START or STOP condition is detected.

43.4.3.2 Transmit and Receive

The transmit and receive data registers are double buffered and only update during a slave-transmit and slave-receive transfer respectively. The slave address that was received can be configured to be read from either the receive data register (for example,

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

when using DMA to transfer data) or from the address status register. The transmit data register can be configured to only request data once a slave-transmit transfer is detected or to request new data whenever the transmit data register is empty.

The transmit data register should only be written when the transmit data flag is set. The receive data register should only be read when the received data flag is set (or the address valid flag is set and RXCFG=1). The address status register should only be read when the address valid flag is set.

43.4.3.3 Clock Stretching

The LPI2C slave supports many configurable options for when clock stretching is performed. The following conditions can be configured to perform clock stretching.

- During 9th clock pulse of address byte and address valid flag is set.
- During 9th clock pulse of slave-transmit transfer and transmit data flag is set.
- During 9th clock pulse of slave-receive transfer and receive data flag is set.
- During 8th clock pulse of address byte or slave-receive transfer and transmit ACK flag is set. This is disabled in high speed mode.
- Clock stretching can also be extended for CLKHOLD cycles to allow additional setup time to sample the SDA pin externally. This is disabled in high speed mode.

Unless extended by the CLKHOLD configuration, clock stretching will extend for one peripheral bus clock cycle after SDA updates when clock stretching is enabled.

43.4.3.4 Timing Parameters

The LPI2C slave can configure the following timing parameters, these parameters are disabled when SCR[FILTEN] is clear, when SCR[FILTDZ] is set in Doze mode, and when LPI2C slave detects high speed mode. When disabled, the LPI2C slave is clocked directly from the I2C bus and may not satisfy all timing requirements of the I2C specification (such as SDA minimum hold time in Standard/Fast mode).

- SDA data valid time from SCL negation to SDA update.
- SCL hold time when clock stretching is enabled to increase setup time when sampling SDA externally.
- SCL glitch filter time.
- SDA glitch filter time.

The LPI2C slave imposes the following restrictions on the timing parameters.

For Assessment Purposes Only

- FILTSDA must be configured to greater than or equal to FILTSCL (unless compensating for board level skew between SDA and SCL).
- DATAVD must be configured less than the minimum SCL low period.

43.4.3.5 Error Conditions

The LPI2C slave can detect the following error conditions.

- Bit error flag will set when the LPI2C slave is driving SDA, but samples a different value than what is expected.
- FIFO error flag will set due to a transmit data underrun or a receive data overrun. Clock stretching can be enabled to eliminate the possibility of underrun and overrun occurring.
- FIFO error flag will also set due to an address overrun when RXCFG is set, otherwise an address overrun is not flagged. Clock stretching can be enabled to eliminate the possibility of overrun occurring.

The LPI2C slave does not implement a timeout due to SCL and/or SDA being stuck low. If this detection is required, the LPI2C master logic should be used and software can reset the LPI2C slave when this condition is detected.

43.4.4 Interrupts and DMA Requests

The LPI2C master and slave interrupts may be combined depending on the device.

The LPI2C master and slave transmit DMA requests may be combined depending on the device.

The LPI2C master and slave receive DMA requests may be combined depending on the device.

43.4.4.1 Master mode

The following table illustrates the master mode sources that can generate the LPI2C master interrupt and LPI2C master transmit/receive DMA requests.

Table 43-6. Master Interrupts and DMA Requests

Flag	Description	Interrupt	DMA Request	Low Power Wakeup
TDF	Data can be written to transmit FIFO, as	Y	TX	Y

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Table 43-6. Master Interrupts and DMA Requests (continued)

Flag	Description	Interrupt	DMA Request	Low Power Wakeup
	configured by TXWATER.			
RDF	Data can be read from the receive FIFO, as configured by RXWATER.	Y	RX	Y
EPF	Master has transmitted Repeated START or STOP condition.	Y	N	Y
SDF	Master has transmitted STOP condition.	Y	N	Y
NDF	Master detected NACK during address byte when expecting ACK, master detected ACK during address byte and expecting NACK, or master detected NACK during master-transmitter data byte.	Y	N	Y
ALF	Master lost arbitration due to START/STOP condition detected at wrong time, or Master was transmitting data but received different data than what was transmitted.	Y	N	Y
FEF	Master expecting START condition in command FIFO and next entry in FIFO is not START condition.	Y	N	Y
PLTF	Pin low timeout is enabled and SCL (or SDA if configured) is low for longer than the configured timeout.	Y	N	Y
DMF	Received data matches the configured data match, and receive data not discarded due to command FIFO entry.	Y	N	Y
MBF	LPI2C master is busy transmitting/receiving data.	N	N	N
BBF	LPI2C master is enabled and activity detected on I2C bus, but STOP condition has not been detected and	N	N	N

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 43-6. Master Interrupts and DMA Requests

Flag	Description	Interrupt	DMA Request	Low Power Wakeup
	bus idle timeout (if enabled) has not occurred.			

43.4.4.2 Slave mode

The following table illustrates the slave mode sources that can generate the LPI2C slave interrupt and the LPI2C slave transmit/receive DMA requests.

Table 43-7. Slave Interrupts and DMA Requests

Flag	Description	Interrupt	DMA Request	Low Power Wakeup
TDF	Data can be written to transmit data register.	Y	TX	Y
RDF	Data can be read from the receive data register.	Y	RX	Y
AVF	Address can be read from the address status register.	Y	RX	Y
TAF	ACK/NACK can be written to the transmit ACK register.	Y	N	Y
RSF	Slave has detected an address match followed by a Repeated START condition.	Y	N	Y
SDF	Slave has detected an address match followed by a STOP condition.	Y	N	Y
BEF	Slave was transmitting data, but received different data than what was transmitted.	Y	N	Y
FEF	Transmit data underrun, receive data overrun or address status overrun (when RXCFG=1). This flag can only set when clock stretching is disabled.	Y	N	Y
AM0F	Slave detected address match with ADDR0 field.	Y	N	N

Table continues on the next page...

s Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 43-7. Slave Interrupts and DMA Requests (continued)

Flag	Description	Interrupt	DMA Request	Low Power Wakeup
AM1F	Slave detected address match with ADDR1 field or address range.	Y	N	N
GCF	Slave detected address match with general call address.	Y	N	N
SARF	Slave detected address match with SMBus alert address.	Y	N	N
SBF	LPI2C slave is busy receiving address byte or transmitting/receiving data.	N	N	N
BBF	LPI2C slave is enabled and START condition detected on I2C bus, but STOP condition has not been detected.	N	N	N

For Assessment Purp

43.4.5 Peripheral Triggers

The connection of the LPI2C peripheral triggers with other peripherals are device specific.

43.4.5.1 Master Output Trigger

The LPI2C master generates an output trigger that can be connected to other peripherals on the device. The master output trigger asserts on both a Repeated START or STOP condition and remains asserted for one cycle of the LPI2C functional clock divided by the prescaler.

43.4.5.2 Slave Output Trigger

The LPI2C slave generates an output trigger that can be connected to other peripherals on the device. The slave output trigger asserts on both a Repeated START or STOP condition that occurs following a slave address match. It remains asserted until the next slave SCL pin negation.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

43.4.5.3 Input Trigger

The LPI2C input trigger can be selected in place of the HREQ pin to control the start of a LPI2C master bus transfer. The input trigger must assert for longer than one LPI2C functional clock cycle to be detected.

43.5 Application Information

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 44
Low Power Universal Asynchronous Receiver/ Transmitter (LPUART)

44.1 Chip-specific Low Power Universal Asynchronous Receiver/Transmitter(LPUART) information

44.1.1 Instantiation Information

The LPUART module supports basic UART with DMA interface function and ×4 to ×32 oversampling of baud-rate.

Table 44-1. LPUART Configuration

LPUART Feature	LPUART0	LPUART1	LPUART2
TX FIFO (word)	4	4	4
RX FIFO (word)	4	4	4

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

This module supports LIN master and slave operation.

44.2 Introduction

44.2.1 Features

Features of the LPUART module include:

- Full-duplex, standard non-return-to-zero (NRZ) format

For Assessment Purposes Only

- 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
  - Supports operation in Stop modes
- 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 7-bit, 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

For Assessment Purposes Only

44.2.2 Modes of operation

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.2.2.1 Stop mode

The LPUART will remain functional during Stop mode, provided the asynchronous transmit and receive clock remains enabled. The LPUART can generate an interrupt or DMA request to cause a wakeup from Stop mode.

44.2.2.2 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.

44.2.2.3 Debug mode

The LPUART remains functional in debug mode.

44.2.3 Signal Descriptions

Signal	Description	I/O
TXD	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
RXD	Receive data.	I
CTS_B	Clear to send.	I
RTS_B	Request to send.	O

44.2.4 Block diagram

The following figure shows the transmitter portion of the LPUART.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

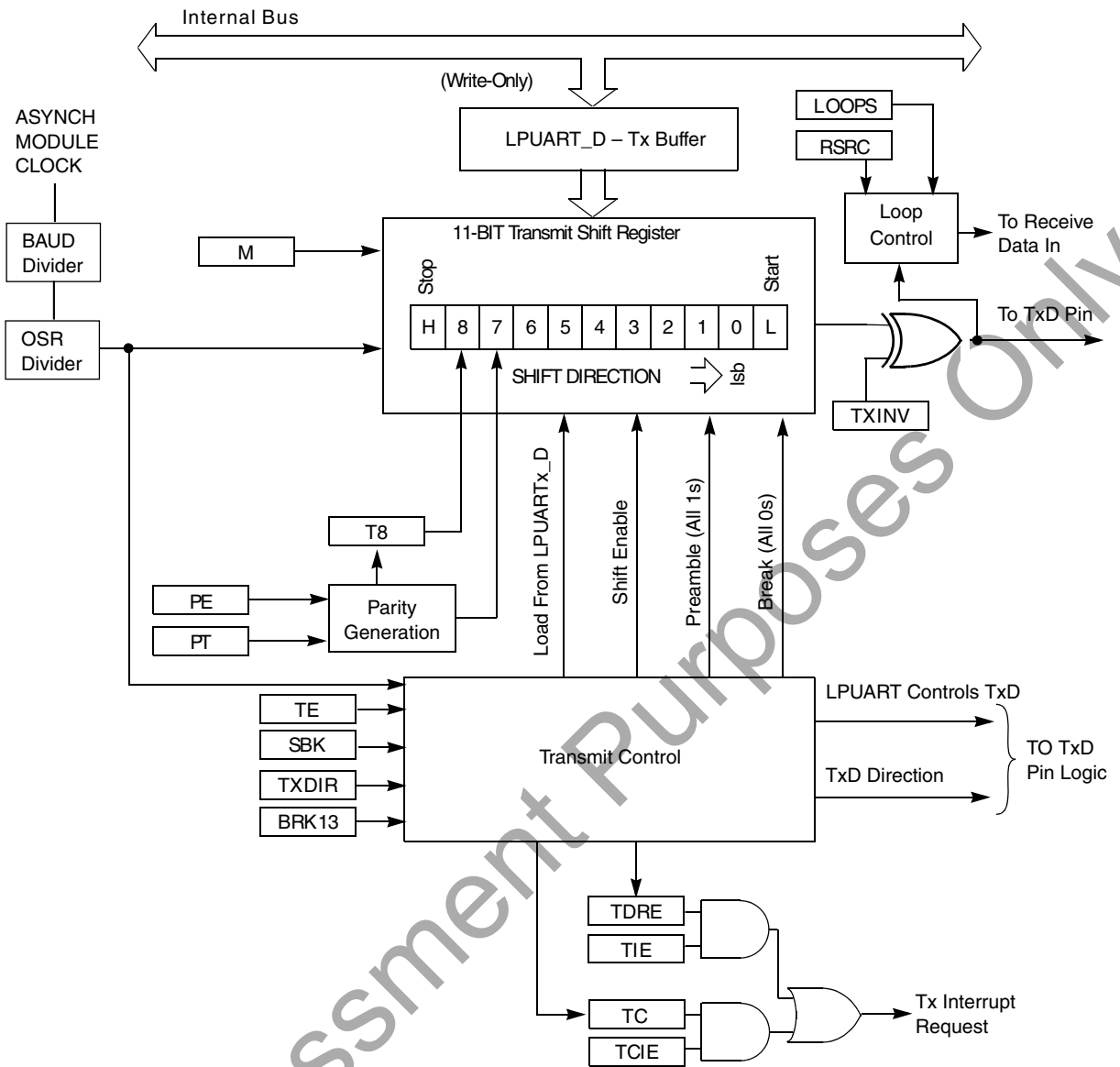


Figure 44-1. LPUART transmitter block diagram

The following figure shows the receiver portion of the LPUART.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

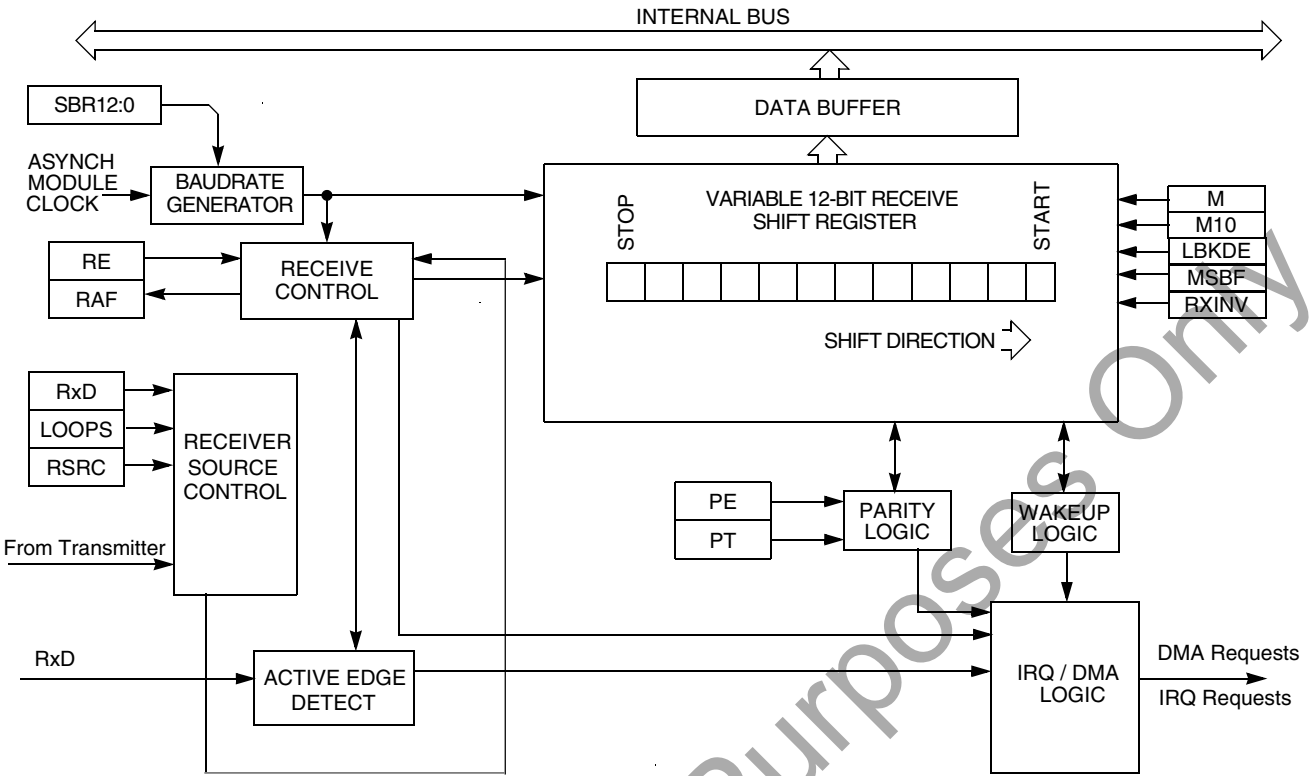


Figure 44-2. LPUART receiver block diagram

44.3 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.

44.3.1 LPUART Register Descriptions

These registers may not be applicable to all instances of LPUART. For more details on the registers supported on each module instance, please refer to "The LPUART as implemented on the chip."

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.3.1.1 LPUART Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID Register (VERID)	32	RO	04010003h
4h	Parameter Register (PARAM)	32	RO	00000202h
8h	LPUART Global Register (GLOBAL)	32	RW	00000000h
Ch	LPUART Pin Configuration Register (PINCFG)	32	RW	00000000h
10h	LPUART Baud Rate Register (BAUD)	32	RW	0F000004h
14h	LPUART Status Register (STAT)	32	RW	00C00000h
18h	LPUART Control Register (CTRL)	32	RW	00000000h
1Ch	LPUART Data Register (DATA)	32	RW	00001000h
20h	LPUART Match Address Register (MATCH)	32	RW	00000000h
24h	LPUART Modem IrDA Register (MODIR)	32	RW	00000000h
28h	LPUART FIFO Register (FIFO)	32	RW	00C00011h
2Ch	LPUART Watermark Register (WATER)	32	RW	00000000h

44.3.1.2 Version ID Register (VERID)

44.3.1.2.1 Address

Register	Offset
VERID	0h

44.3.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	1	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	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.3.1.2.3 Fields

Field	Function
31-24 MAJOR	Major Version Number This read only field returns the major version number for the module specification.
23-16 MINOR	Minor Version Number This read only field returns the minor version number for the module specification.
15-0 FEATURE	Feature Identification Number This read only field returns the feature set number. 0000000000000001b - Standard feature set. 0000000000000011b - Standard feature set with MODEM/IrDA support.

44.3.1.3 Parameter Register (PARAM)

44.3.1.3.1 Address

Register	Offset
PARAM	4h

44.3.1.3.2 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	RXFIFO								TXFIFO							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

44.3.1.3.3 Fields

Field	Function
31-16 —	Reserved
15-8	Receive FIFO Size

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
RXFIFO	The number of words in the receive FIFO is 2^RXFIFO.
7-0 TXFIFO	Transmit FIFO Size The number of words in the transmit FIFO is 2^TXFIFO.

44.3.1.4 LPUART Global Register (GLOBAL)

44.3.1.4.1 Address

Register	Offset
GLOBAL	8h

44.3.1.4.2 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	0														RST	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

44.3.1.4.3 Fields

Field	Function
31-2 Reserved	Reserved
1 RST	Software Reset Reset all internal logic and registers, except the Global Register. Remains set until cleared by software. 0b - Module is not reset. 1b - Module is reset.
0 —	Reserved

Freescale Confidential Proprietary Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.3.1.5 LPUART Pin Configuration Register (PINCFG)

44.3.1.5.1 Address

Register	Offset
PINCFG	Ch

44.3.1.5.2 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														TRGSEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

44.3.1.5.3 Fields

Field	Function
31-2 Reserved	Reserved
1-0 TRGSEL	Trigger Select Configures the input trigger usage. 00b - Input trigger is disabled. 01b - Input trigger is used instead of RXD pin input. 10b - Input trigger is used instead of CTS_B pin input. 11b - Input trigger is used to modulate the TXD pin output.

44.3.1.6 LPUART Baud Rate Register (BAUD)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.3.1.6.1 Address

Register	Offset
BAUD	10h

44.3.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAEN1	MAEN2	M10	OSR					TDMAE	0	RDMAE	RIDMAE	MATCFG		BOTHEDGE	RESYNCDIS
W																
Reset	0	0	0	0	1	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	LBKDIE	RXEDGIE	SBNS	SBR												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

44.3.1.6.3 Fields

Field	Function
31 MAEN1	Match Address Mode Enable 1 0b - Normal operation. 1b - Enables automatic address matching or data matching mode for MATCH[MA1].
30 MAEN2	Match Address Mode Enable 2 0b - Normal operation. 1b - 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. 0b - Receiver and transmitter use 7-bit to 9-bit data characters. 1b - 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 (i.e., a value not between 4x and 32x) will default to an oversampling ratio of 16 (01111). This field should only be changed when the transmitter and receiver are both disabled.
23 TDMAE	Transmitter DMA Enable TDMAE configures the transmit data register empty flag, LPUART_STAT[TDRE], to generate a DMA request. 0b - DMA request disabled. 1b - DMA request enabled.
22 —	Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
21 RDMAE	Receiver Full DMA Enable RDMAE configures the receiver data register full flag, LPUART_STAT[RDRF], to generate a DMA request. 0b - DMA request disabled. 1b - DMA request enabled.
20 RIDMAE	Receiver Idle DMA Enable RIDMAE configures the receiver idle flag, LPUART_STAT[IDLE], to generate a DMA request. When this bit is set, reading LPUART_DATA when either DATA[RXEMPT] or DATA[IDLINE] bit is set, will generate an End Of Packet response until the completion of the existing DMA transfer. During an End of Packet response, reading the LPUART_DATA register will return 0x0000_33FF and does not pull data from the FIFO. 0b - DMA request disabled. 1b - DMA request enabled.
19-18 MATCFG	Match Configuration Configures the match addressing mode used. 00b - Address Match Wakeup 01b - Idle Match Wakeup 10b - Match On and Match Off 11b - 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. 0b - Receiver samples input data using the rising edge of the baud rate clock. 1b - 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. 0b - Resynchronization during received data word is supported 1b - Resynchronization during received data word is disabled
15 LBKDIE	LIN Break Detect Interrupt Enable LBKDIE enables the LIN break detect flag, LBKDIF, to generate interrupt requests. 0b - Hardware interrupts from LPUART_STAT[LBKDIF] disabled (use polling). 1b - 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. 0b - Hardware interrupts from LPUART_STAT[RXEDGIF] disabled (use polling). 1b - 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. 0b - One stop bit. 1b - Two stop bits.
12-0 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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	only be updated when the transmitter and receiver are both disabled (LPUART_CTRL[RE] and LPUART_CTRL[TE] are both 0).

44.3.1.7 LPUART Status Register (STAT)

44.3.1.7.1 Address

Register	Offset
STAT	14h

44.3.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W	w1c	w1c										w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MA1F	MA2F	0													
W	w1c	w1c														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

44.3.1.7.3 Fields

Field	Function
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. 0b - No LIN break character has been detected. 1b - LIN break character has been detected.
30 RXEDGIF	RXD Pin Active Edge Interrupt Flag RXEDGIF is set when an active edge, falling if RXINV = 0, rising if RXINV=1, on the RXD pin occurs. RXEDGIF is cleared by writing a 1 to it. 0b - No active edge on the receive pin has occurred. 1b - An active edge on the receive pin has occurred.
29	MSB First

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
MSBF	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. 0b - 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. 1b - 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 RXD input for all cases: data bits, start and stop bits, break, and idle. 0b - Receive data not inverted. 1b - 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. 0b - 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. 1b - 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. 0b - Break character is transmitted with length of 9 to 13 bit times. 1b - Break character is transmitted with length of 12 to 15 bit times.
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. 0b - LIN break detect is disabled, normal break character can be detected. 1b - LIN break detect is enabled. LIN break character is detected at length of 11 bit times (if M = 0) or 12 (if M = 1) or 13 (M10 = 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. 0b - LPUART receiver idle waiting for a start bit. 1b - LPUART receiver active (RXD 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. 0b - Transmit data buffer full. 1b - Transmit data buffer empty.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
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]. 0b - Transmitter active (sending data, a preamble, or a break). 1b - 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. 0b - Receive data buffer empty. 1b - 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. 0b - No idle line detected. 1b - Idle line was detected.
19 OR	Receiver Overrun Flag OR is set when software fails to prevent the receive data register from overflowing with data. The OR bit is set immediately after the stop bit has been completely received for the dataword that overflows the buffer and all the other error flags (FE, NF, and PF) are prevented from setting. The data in the shift register is lost, but the data already in the LPUART data registers is not affected. If LBKDE is enabled and a LIN Break is detected, the OR field asserts if LBKDIF is not cleared before the next data character is received. While the OR flag is set, no additional data is stored in the data buffer even if sufficient room exists. To clear OR, write logic 1 to the OR flag. 0b - No overrun. 1b - 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. 0b - No noise detected. 1b - Noise detected in the received character in LPUART_DATA.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
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. 0b - No framing error detected. This does not guarantee the framing is correct. 1b - 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. 0b - No parity error. 1b - 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. 0b - Received data is not equal to MA1 1b - 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. 0b - Received data is not equal to MA2 1b - Received data is equal to MA2
13-0 —	Reserved

44.3.1.8 LPUART Control Register (CTRL)

44.3.1.8.1 Address

Register	Offset
CTRL	18h

44.3.1.8.2 Function

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.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.3.1.8.3 Diagram

Bits	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

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MA1IE	MA2IE	0		M7	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

44.3.1.8.4 Fields

Field	Function
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	TXD 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 TXD pin. When clearing TXDIR, the transmitter will finish receiving the current character (if any) before the receiver starts receiving data from the TXD pin. 0b - TXD pin is an input in single-wire mode. 1b - TXD pin is an output in single-wire mode.
28 TXINV	Transmit Data Inversion Setting this bit reverses the polarity of the transmitted data output. **NOTE:** Setting TXINV inverts the TXD output for all cases: data bits, start and stop bits, break, and idle. 0b - Transmit data not inverted. 1b - Transmit data inverted.
27 ORIE	Overrun Interrupt Enable This bit enables the overrun flag (OR) to generate hardware interrupt requests. 0b - OR interrupts disabled; use polling. 1b - Hardware interrupt requested when OR is set.
26	Noise Error Interrupt Enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
NEIE	This bit enables the noise flag (NF) to generate hardware interrupt requests. 0b - NF interrupts disabled; use polling. 1b - 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. 0b - FE interrupts disabled; use polling. 1b - 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. 0b - PF interrupts disabled; use polling). 1b - Hardware interrupt requested when PF is set.
23 TIE	Transmit Interrupt Enable Enables STAT[TDRE] to generate interrupt requests. 0b - Hardware interrupts from TDRE disabled; use polling. 1b - 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. 0b - Hardware interrupts from TC disabled; use polling. 1b - Hardware interrupt requested when TC flag is 1.
21 RIE	Receiver Interrupt Enable Enables STAT[RDRF] to generate interrupt requests. 0b - Hardware interrupts from RDRF disabled; use polling. 1b - Hardware interrupt requested when RDRF flag is 1.
20 ILIE	Idle Line Interrupt Enable ILIE enables the idle line flag, STAT[IDLE], to generate interrupt requests. 0b - Hardware interrupts from IDLE disabled; use polling. 1b - 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 TXD pin is tristated. 0b - Transmitter disabled. 1b - 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). 0b - Receiver disabled. 1b - 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. 0b - Normal receiver operation.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	1b - 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 9 to 13 bits, or 12 to 15 bits if LPUART_STAT[BRK13] 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. 0b - Normal transmitter operation. 1b - Queue break character(s) to be sent.
15 MA1IE	Match 1 Interrupt Enable 0b - MA1F interrupt disabled 1b - MA1F interrupt enabled
14 MA2IE	Match 2 Interrupt Enable 0b - MA2F interrupt disabled 1b - MA2F interrupt enabled
13-12 —	Reserved
11 M7	7-Bit Mode Select This bit should only be changed when the transmitter and receiver are both disabled. 0b - Receiver and transmitter use 8-bit to 10-bit data characters. 1b - Receiver and transmitter use 7-bit data characters.
10-8 IDLECFG	Idle Configuration Configures the number of idle characters that must be received before the IDLE flag is set. 000b - 1 idle character 001b - 2 idle characters 010b - 4 idle characters 011b - 8 idle characters 100b - 16 idle characters 101b - 32 idle characters 110b - 64 idle characters 111b - 128 idle characters
7 LOOPS	Loop Mode Select When LOOPS is set, the RXD 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. 0b - Normal operation - RXD and TXD use separate pins. 1b - Loop mode or single-wire mode where transmitter outputs are internally connected to receiver input (see RSRC bit).
6 DOZEEN	Doze Enable 0b - LPUART is enabled in Doze mode. 1b - 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. 0b - Provided LOOPS is set, RSRC is cleared, selects internal loop back mode and the LPUART does not use the RXD pin. 1b - Single-wire LPUART mode where the TXD pin is connected to the transmitter output and receiver input.
4 M	9-Bit or 8-Bit Mode Select 0b - Receiver and transmitter use 8-bit data characters. 1b - Receiver and transmitter use 9-bit data characters.

Table continues on the next page...

Review Draft
Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
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. 0b - Configures RWU for idle-line wakeup. 1b - Configures RWU with address-mark wakeup.
2 ILT	Idle Line Type Select Determines when the receiver starts counting logic 1s as idle character bits. The count begins either after a valid start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit can cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. **NOTE:** In case the LPUART is programmed with ILT = 1, a logic 0 is automatically shifted after a received stop bit, therefore resetting the idle count. 0b - Idle character bit count starts after start bit. 1b - 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. 0b - No hardware parity generation or checking. 1b - 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. 0b - Even parity. 1b - Odd parity.

44.3.1.9 LPUART Data Register (DATA)

44.3.1.9.1 Address

Register	Offset
DATA	1Ch

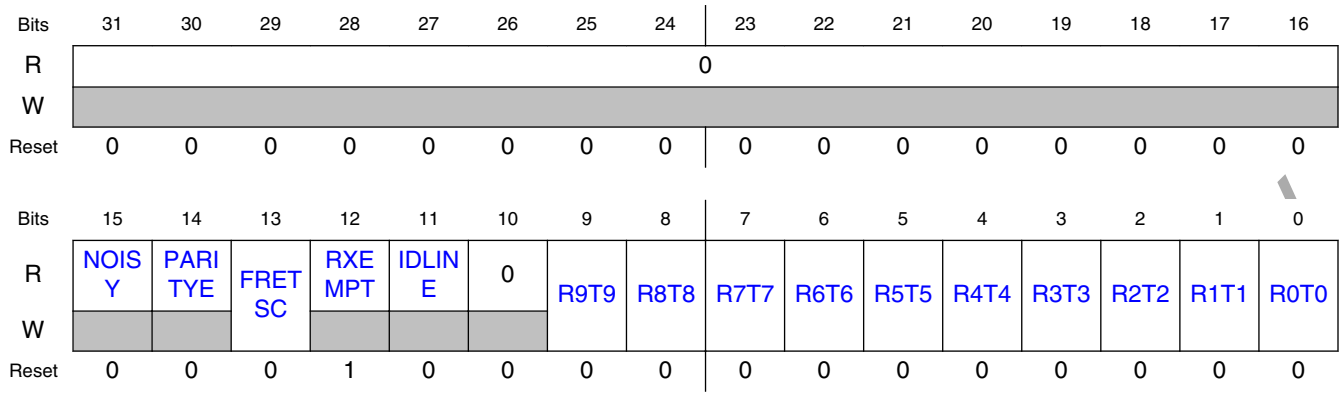
44.3.1.9.2 Function

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.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.3.1.9.3 Diagram



44.3.1.9.4 Fields

Field	Function
31-16 —	Reserved
15 NOISY	NOISY The current received dataword contained in DATA[R9:R0] was received with noise. 0b - The dataword was received without noise. 1b - The data was received with noise.
14 PARITYE	PARITYE The current received dataword contained in DATA[R9:R0] was received with a parity error. 0b - The dataword was received without a parity error. 1b - 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. 0b - The dataword was received without a frame error on read, transmit a normal character on write. 1b - 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. 0b - Receive buffer contains valid data. 1b - 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. 0b - Receiver was not idle before receiving this character. 1b - Receiver was idle before receiving this character.
10 —	Reserved

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
9 R9T9	R9T9 Read receive data buffer 9 or write transmit data buffer 9.
8 R8T8	R8T8 Read receive data buffer 8 or write transmit data buffer 8.
7 R7T7	R7T7 Read receive data buffer 7 or write transmit data buffer 7.
6 R6T6	R6T6 Read receive data buffer 6 or write transmit data buffer 6.
5 R5T5	R5T5 Read receive data buffer 5 or write transmit data buffer 5.
4 R4T4	R4T4 Read receive data buffer 4 or write transmit data buffer 4.
3 R3T3	R3T3 Read receive data buffer 3 or write transmit data buffer 3.
2 R2T2	R2T2 Read receive data buffer 2 or write transmit data buffer 2.
1 R1T1	R1T1 Read receive data buffer 1 or write transmit data buffer 1.
0 R0T0	R0T0 Read receive data buffer 0 or write transmit data buffer 0.

44.3.1.10 LPUART Match Address Register (MATCH)

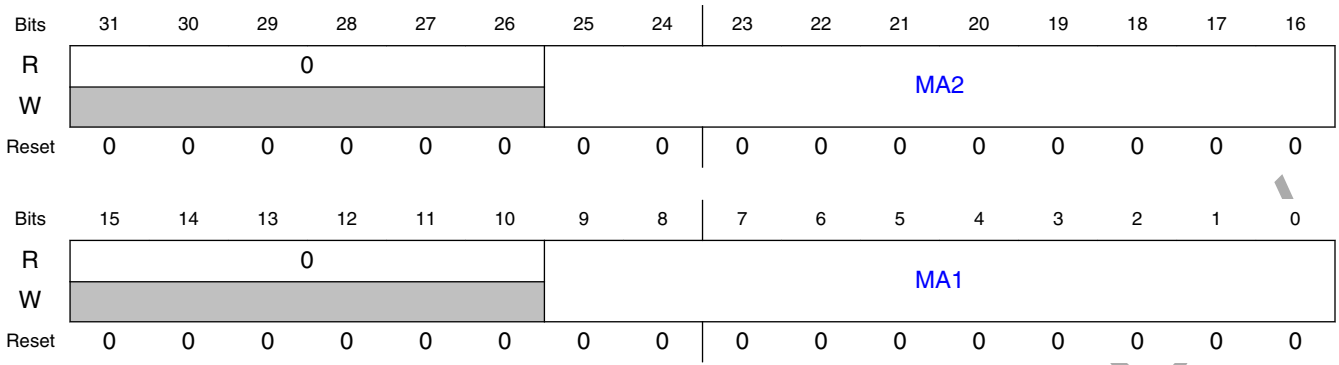
44.3.1.10.1 Address

Register	Offset
MATCH	20h

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.3.1.10.2 Diagram



44.3.1.10.3 Fields

Field	Function
31-26 —	Reserved
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
9-0 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.

44.3.1.11 LPUART Modem IrDA Register (MODIR)

44.3.1.11.1 Address

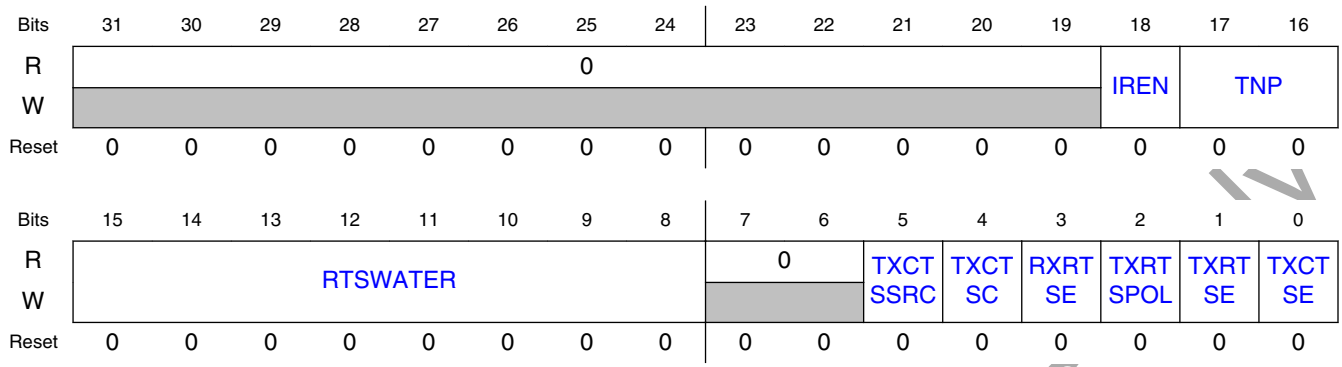
Register	Offset
MODIR	24h

44.3.1.11.2 Function

The MODEM register controls options for setting the modem configuration.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.3.1.11.3 Diagram



44.3.1.11.4 Fields

Field	Function
31-19 —	Reserved
18 IREN	Infrared enable Enables/disables the infrared modulation/demodulation. 0b - IR disabled. 1b - 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. 00b - 1/OSR. 01b - 2/OSR. 10b - 3/OSR. 11b - 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. 00000000b - RTS asserts when the receiver FIFO is full or receiving a character that causes the FIFO to become full. 00000001b - 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
5 TXCTSSRC	Transmit CTS Source Configures the source of the CTS input. 0b - CTS input is the CTS_B pin. 1b - 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. 0b - CTS input is sampled at the start of each character.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	1b - 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. 0b - The receiver has no effect on RTS. 1b - RTS is deasserted if the receiver data register is full or a start bit has been detected that would cause the receiver data register to become full. RTS is asserted if the receiver data register is not full and has not detected a start bit that would cause the receiver data register to become full.
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. 0b - Transmitter RTS is active low. 1b - Transmitter RTS is active high.
1 TXRTSE	Transmitter request-to-send enable Controls RTS before and after a transmission. 0b - The transmitter has no effect on RTS. 1b - 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. 0b - CTS has no effect on the transmitter. 1b - 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.

44.3.1.12 LPUART FIFO Register (FIFO)

44.3.1.12.1 Address

Register	Offset
FIFO	28h

44.3.1.12.2 Function

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.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.3.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								TXEMPT	RXEMPT	0				TXOF	RXUF
W															w1c	w1c
Reset	0	0	0	0	0	0	0	0	1	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	0	0	0	RXIDEN			TXOFE	RXUFE	TXFE	TXFIFOSIZE			RXFE	RXFIFOSIZE		
W	TXFLUSH	RXFLUSH														
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

44.3.1.12.4 Fields

Field	Function
31-24 —	Reserved
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. 0b - Transmit buffer is not empty. 1b - Transmit buffer is empty.
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. 0b - Receive buffer is not empty. 1b - Receive buffer is empty.
21-18 —	Reserved
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. 0b - No transmit buffer overflow has occurred since the last time the flag was cleared. 1b - 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. 0b - No receive buffer underflow has occurred since the last time the flag was cleared. 1b - At least one receive buffer underflow has occurred since the last time the flag was cleared.
15 TXFLUSH	Transmit FIFO/Buffer Flush

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	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. 0b - No flush operation occurs. 1b - 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. 0b - No flush operation occurs. 1b - All data in the receive FIFO/buffer is cleared out.
13 —	Reserved
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. 000b - Disable RDRF assertion due to partially filled FIFO when receiver is idle. 001b - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 1 character. 010b - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 2 characters. 011b - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 4 characters. 100b - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 8 characters. 101b - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 16 characters. 110b - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 32 characters. 111b - 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. 0b - TXOF flag does not generate an interrupt to the host. 1b - 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. 0b - RXUF flag does not generate an interrupt to the host. 1b - 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. 0b - Transmit FIFO is not enabled. Buffer is depth 1. (Legacy support). 1b - 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. 000b - Transmit FIFO/Buffer depth = 1 dataword. 001b - Transmit FIFO/Buffer depth = 4 datawords. 010b - Transmit FIFO/Buffer depth = 8 datawords. 011b - Transmit FIFO/Buffer depth = 16 datawords. 100b - Transmit FIFO/Buffer depth = 32 datawords. 101b - Transmit FIFO/Buffer depth = 64 datawords. 110b - Transmit FIFO/Buffer depth = 128 datawords. 111b - Transmit FIFO/Buffer depth = 256 datawords
3 RXFE	Receive FIFO Enable

Table continues on the next page...

Review Draft
Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	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. 0b - Receive FIFO is not enabled. Buffer is depth 1. (Legacy support) 1b - Receive FIFO is enabled. Buffer is depth indicted by RXFIFOSIZE.
2-0 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. 000b - Receive FIFO/Buffer depth = 1 dataword. 001b - Receive FIFO/Buffer depth = 4 datawords. 010b - Receive FIFO/Buffer depth = 8 datawords. 011b - Receive FIFO/Buffer depth = 16 datawords. 100b - Receive FIFO/Buffer depth = 32 datawords. 101b - Receive FIFO/Buffer depth = 64 datawords. 110b - Receive FIFO/Buffer depth = 128 datawords. 111b - Receive FIFO/Buffer depth = 256 datawords.

44.3.1.13 LPUART Watermark Register (WATER)

44.3.1.13.1 Address

Register	Offset
WATER	2Ch

44.3.1.13.2 Function

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.

44.3.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXCOUNT								RXWATER							
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	TXCOUNT								TXWATER							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.3.1.13.4 Fields

Field	Function
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.
7-0 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].

44.4 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.

44.4.1 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.

For Assessment

Freescale Confidential Proprietary - Non-Disclosure Agreement required

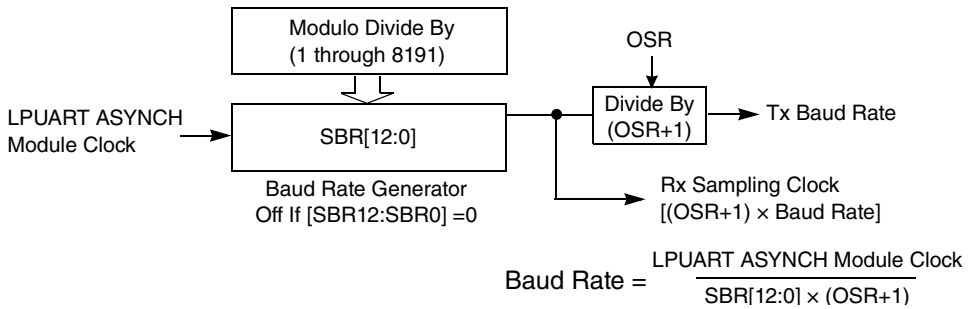


Figure 44-3. LPUART baud rate generation

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.

44.4.2 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 (TXD) 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 9-bit to 13 bits long depending on the setting in the CTRL[M], CTRL[M7], BAUD[M10] and BAUD[SBNS] control bits. For the remainder of this section, assume CTRL[M], CTRL[M7], 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 TXD pin, the transmitter sets the transmit complete flag and enters an idle mode, with TXD high, waiting for more characters to transmit.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

44.4.2.1 Hardware flow control

The transmitter supports hardware flow control by gating the transmission with the value of CTS_B. If the clear-to-send operation is enabled, the character is transmitted when CTS_B is asserted. If CTS_B 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 TXD remains in the mark state until CTS_B is reasserted.

If the clear-to-send operation is disabled, the transmitter ignores the state of CTS_B.

The transmitter's CTS_B signal can also be enabled even if the same LPUART receiver's RTS_B signal is disabled.

44.4.2.2 Transceiver driver enable

The transmitter can use RTS_B as an enable signal for the driver of an external transceiver. See Transceiver driver enable using RTS_B for details. If the request-to-send operation is enabled, when a character is placed into an empty transmitter data buffer, RTS_B asserts one bit time before the start bit is transmitted. RTS_B remains asserted for the whole time that the transmitter data buffer has any characters. RTS_B 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 RTS_B, with the same assertion and deassertion timing as having a character in the transmitter data buffer.

The transmitter's RTS_B signal asserts only when the transmitter is enabled. However, the transmitter's RTS_B signal is unaffected by its CTS_B signal. RTS_B will remain asserted until the transfer is completed, even if the transmitter is disabled mid-way through a data transfer.

For Assessment Purposes Only

44.4.3 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, seven to ten data bits (msb or lsb first), and one or two stop bits of logic 1. For information about 7-bit, 9-bit or 10-bit data mode, refer to 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.

44.4.3.1 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 RXD 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

44.4.3.2 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 44-2. 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

For Assessment

44.4.3.2.1 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], LPUART_CTRL[M7] and LPUART_BAUD[M10] control bit selects 7-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, 9 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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

44.4.3.2.2 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.

44.4.3.2.3 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.

44.4.3.2.4 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 RXD 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

44.4.3.2.5 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 RXD 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.

44.4.3.2.6 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 RXD 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

For Assessment Purposes Only

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.

44.4.3.3 Hardware flow control

To support hardware flow control, the receiver can be programmed to automatically deassert and assert RTS_B.

- RTS_B remains asserted until the transfer is complete, even if the transmitter is disabled midway through a data transfer. See Transceiver driver enable using RTS_B for more details.
- If the receiver request-to-send functionality is enabled, the receiver automatically deasserts RTS_B 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 RTS_B 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 RTS_B 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 RTS_B remains deasserted.

44.4.3.4 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.

44.4.3.4.1 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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

44.4.3.4.2 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.

44.4.3.4.3 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.

44.4.3.4.4 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.

44.4.4 Additional LPUART functions

The following sections describe additional LPUART functions.

44.4.4.1 Data Modes

The LPUART transmitter and receiver can be configured to operate in 7-bit data mode by setting LPUART_CTRL[M7], 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 Assessment Purposes Only

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.

44.4.4.2 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.

44.4.4.3 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,

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 RXD pin is not used by the LPUART.

44.4.4.4 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 TXD pin (the RXD pin is not used).

In single-wire mode, the LPUART_CTRL[TXDIR] bit controls the direction of serial data on the TXD pin. When LPUART_CTRL[TXDIR] is cleared, the TXD pin is an input to the receiver and the transmitter is temporarily disconnected from the TXD pin so an external device can send serial data to the receiver. When LPUART_CTRL[TXDIR] is set, the TXD 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.

44.4.5 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

44.4.5.1 Infrared transmit encoder

The infrared transmit encoder converts serial bits of data from transmit shift register to the TXD 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.

44.4.5.2 Infrared receive decoder

The infrared receive block converts data from the RXD 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.

44.4.6 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 TXD 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.

For Assessment Purposes Only

The IDLE status flag includes logic that prevents it from getting set repeatedly when the RXD 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 RXD 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).

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 45
Flexible I/O (FlexIO)

45.1 Chip-specific Flexible I/O (FlexIO) information

45.1.1 FlexIO Configuration

Table 45-1. FlexIO Configuration

	Timers	Shifters	Pins
Number	4	4	8

45.2 Introduction

45.2.1 Overview

The FlexIO is a highly configurable module providing a wide range of functionality including:

- Emulation of a variety of serial communication protocols
- Flexible 16-bit timers with support for a variety of trigger, reset, enable and disable conditions

45.2.2 Features

The FlexIO module is capable of supporting a wide range of protocols including, but not limited to:

- UART
- I2C

For Assessment Purposes Only

- SPI
- I2S
- PWM/Waveform generation

The following key features are provided:

- Array of 32-bit shift registers with transmit, receive and data match modes
- Double buffered shifter operation for continuous data transfer
- Shifter concatenation to support large transfer sizes
- Automatic start/stop bit generation
- Interrupt, DMA or polled transmit/receive operation
- Programmable baud rates independent of bus clock frequency, with support for asynchronous operation during stop modes
- Highly flexible 16-bit timers with support for a variety of internal or external trigger, reset, enable and disable conditions

45.2.3 Block Diagram

The following diagram gives a high-level overview of the configuration of FlexIO timers and shifters.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

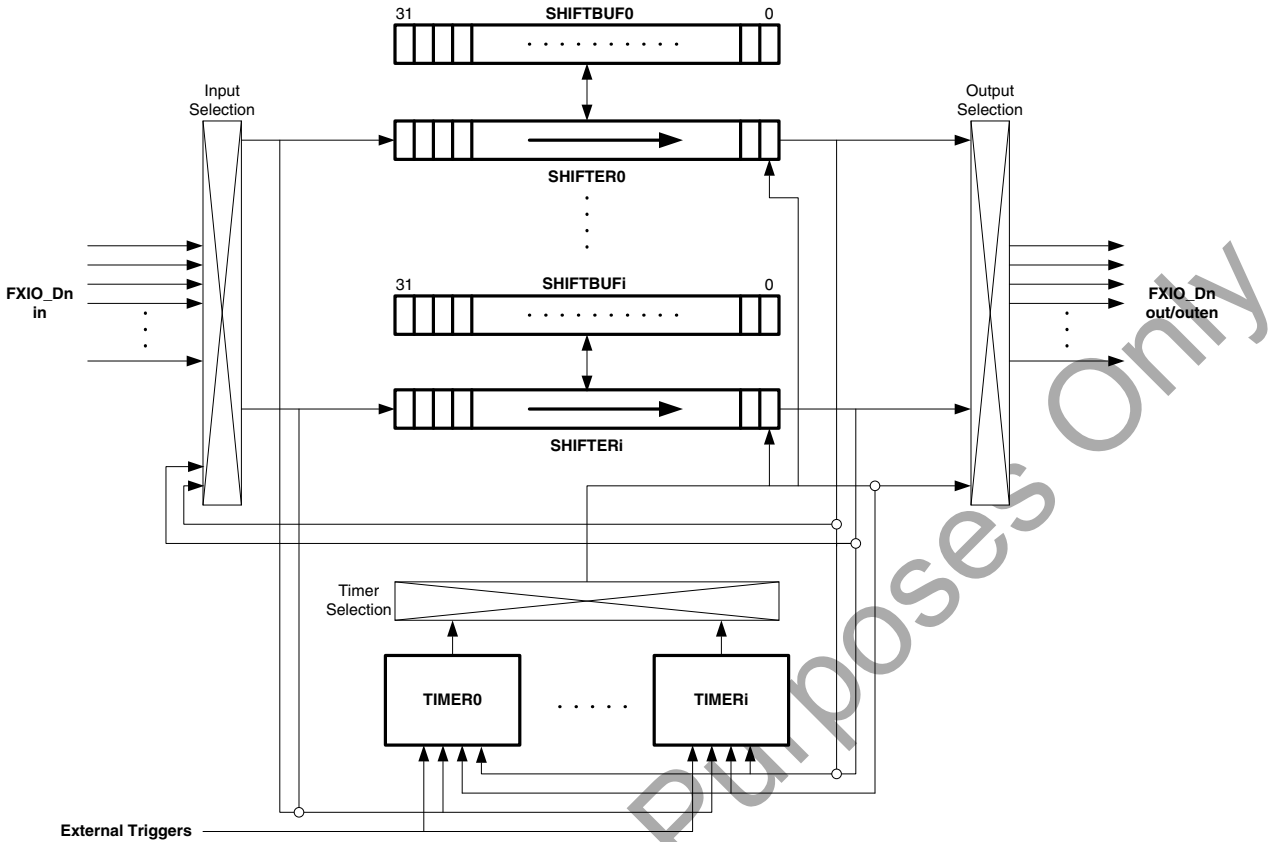


Figure 45-1. FlexIO block diagram

45.2.4 Modes of operation

The FlexIO module supports the chip modes described in the following table.

Table 45-2. Chip modes supported by the FlexIO module

Chip mode	FlexIO Operation
Run	Normal operation
Stop/Wait	Can continue operating provided the Doze Enable bit (CTRL[DOZEN]) is clear and the FlexIO is using an external or internal clock source which remains operating during stop/wait modes.
Low Leakage Stop	The Doze Enable (CTRL[DOZEN]) bit is ignored and the FlexIO will wait for all Timers to complete any pending operation before acknowledging low leakage mode entry.
Debug	Can continue operating provided the Debug Enable bit (CTRL[DBGE]) is set.

Assignment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.2.5 FlexIO Signal Descriptions

Signal	Description	I/O
FXIO_Dn (n=0...7)	Bidirectional FlexIO Shifter and Timer pin inputs/outputs	I/O

45.3 Memory Map and Registers

45.3.1 FLEXIO Register Descriptions

45.3.1.1 FLEXIO Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID Register (VERID)	32	RO	01010001h
4h	Parameter Register (PARAM)	32	RO	10080404h
8h	FlexIO Control Register (CTRL)	32	RW	00000000h
Ch	Pin State Register (PIN)	32	RO	00000000h
10h	Shifter Status Register (SHIFTSTAT)	32	W1C	00000000h
14h	Shifter Error Register (SHIFTERR)	32	W1C	00000000h
18h	Timer Status Register (TIMSTAT)	32	W1C	00000000h
20h	Shifter Status Interrupt Enable (SHIFTSIEN)	32	RW	00000000h
24h	Shifter Error Interrupt Enable (SHIFTEIEN)	32	RW	00000000h
28h	Timer Interrupt Enable Register (TIMIEN)	32	RW	00000000h
30h	Shifter Status DMA Enable (SHIFTSDEN)	32	RW	00000000h
80h - 8Ch	Shifter Control N (SHIFTCTL0 - SHIFTCTL3)	32	RW	00000000h
100h - 10Ch	Shifter Configuration N (SHIFTCFG0 - SHIFTCFG3)	32	RW	00000000h
200h - 20Ch	Shifter Buffer N (SHIFTBUF0 - SHIFTBUF3)	32	RW	00000000h
280h - 28Ch	Shifter Buffer N Bit Swapped (SHIFTBUFBIS0 - SHIFTBUFBIS3)	32	RW	00000000h
300h - 30Ch	Shifter Buffer N Byte Swapped (SHIFTBUFBYS0 - SHIFTBUFBYS3)	32	RW	00000000h
380h - 38Ch	Shifter Buffer N Bit Byte Swapped (SHIFTBUFBBS0 - SHIFTBUFBBS3)	32	RW	00000000h
400h - 40Ch	Timer Control N (TIMCTL0 - TIMCTL3)	32	RW	00000000h
480h - 48Ch	Timer Configuration N (TIMCFG0 - TIMCFG3)	32	RW	00000000h
500h - 50Ch	Timer Compare N (TIMCMP0 - TIMCMP3)	32	RW	00000000h

rposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.2 Version ID Register (VERID)

45.3.1.2.1 Address

Register	Offset
VERID	0h

45.3.1.2.2 Function

.

45.3.1.2.3 Diagram

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

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

45.3.1.2.4 Fields

Field	Function
0-15 FEATURE	Feature Specification Number This read only field returns the feature set number. 0000000000000000b - Standard features implemented. 0000000000000001b - Supports state, logic and parallel modes.
16-23 MINOR	Minor Version Number This read only field returns the minor version number for the module specification.
24-31 MAJOR	Major Version Number This read only field returns the major version number for the module specification.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.3 Parameter Register (PARAM)

45.3.1.3.1 Address

Register	Offset
PARAM	4h

45.3.1.3.2 Function

.

45.3.1.3.3 Diagram

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

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

45.3.1.3.4 Fields

Field	Function
0-7 SHIFTER	Shifter Number Number of Shifters implemented.
8-15 TIMER	Timer Number Number of Timers implemented.
16-23 PIN	Pin Number Number of Pins implemented.
24-31 TRIGGER	Trigger Number Number of external triggers implemented.

45.3.1.4 FlexIO Control Register (CTRL)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

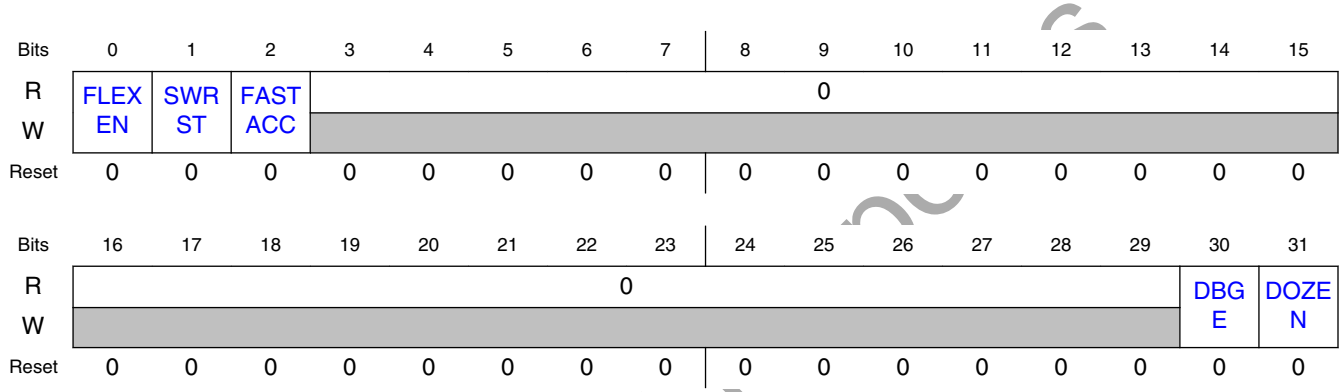
45.3.1.4.1 Address

Register	Offset
CTRL	8h

45.3.1.4.2 Function

.

45.3.1.4.3 Diagram



45.3.1.4.4 Fields

Field	Function
0 FLEXEN	FlexIO Enable 0b - FlexIO module is disabled. 1b - FlexIO module is enabled.
1 SWRST	Software Reset The FlexIO Control Register is not affected by the software reset, all other logic in the FlexIO is affected by the software reset and register accesses are ignored until this bit is cleared. This register bit will remain set until cleared by software, and the reset has cleared in the FlexIO clock domain. 0b - Software reset is disabled 1b - Software reset is enabled, all FlexIO registers except the Control Register are reset.
2 FASTACC	Fast Access Enables fast register accesses to FlexIO registers, but requires the FlexIO clock to be at least twice the frequency of the bus clock. 0b - Configures for normal register accesses to FlexIO 1b - Configures for fast register accesses to FlexIO
3-29 Reserved	Reserved.
30 DBGE	Debug Enable Enables FlexIO operation in Debug mode. 0b - FlexIO is disabled in debug modes.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	1b - FlexIO is enabled in debug modes
31 DOZEN	Doze Enable Disables FlexIO operation in Doze modes. This field is ignored and the FlexIO always disabled in low-leakage stop modes. 0b - FlexIO enabled in Doze modes. 1b - FlexIO disabled in Doze modes.

45.3.1.5 Pin State Register (PIN)

45.3.1.5.1 Address

Register	Offset
PIN	Ch

45.3.1.5.2 Diagram

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

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

45.3.1.5.3 Fields

Field	Function
0-7 PDI	Pin Data Input Returns the input data on each of the FlexIO pins.
8-31 Reserved	Reserved.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.6 Shifter Status Register (SHIFTSTAT)

45.3.1.6.1 Address

Register	Offset
SHIFTSTAT	10h

45.3.1.6.2 Function

.

45.3.1.6.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	SSF				0											
W	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

45.3.1.6.4 Fields

Field	Function
0-3	Shifter Status Flag
SSF	The shifter status flag is updated when one of the following events occurs: For SMOD=Receive, the status flag is set when SHIFTBUF has been loaded with data from Shifter (SHIFTBUF is full), and the status flag is cleared when SHIFTBUF register is read. For SMOD=Transmit, the status flag is set when SHIFTBUF data has been transferred to the Shifter (SHIFTBUF is empty) or when initially configured for SMOD=Transmit, and the status flag is cleared when the SHIFTBUF register is written. For SMOD=Match Store, the status flag is set when a match has occured between SHIFTBUF and Shifter, and the status flag is cleared when the SHIFTBUF register is read. For SMOD=Match Continuous, returns the current match result between the SHIFTBUF and Shifter. The status flag can also be cleared by writing a logic one to the flag for all modes except Match Continuous. 0000b - Status flag is clear 0001b - Status flag is set
4-31	Reserved.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
Reserved	

45.3.1.7 Shifter Error Register (SHIFTERR)

45.3.1.7.1 Address

Register	Offset
SHIFTERR	14h

45.3.1.7.2 Function

.

45.3.1.7.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	SEF				0											
W	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

45.3.1.7.4 Fields

Field	Function
0-3	Shifter Error Flags
SEF	The shifter error flag is set when one of the following events occurs: For SMOD=Receive, indicates Shifter was ready to store new data into SHIFTBUF before the previous data was read from SHIFTBUF (SHIFTBUF Overrun), or indicates that the received start or stop bit does not match the expected value. For SMOD=Transmit, indicates Shifter was ready to load new data from SHIFTBUF before new data had been written into SHIFTBUF (SHIFTBUF Underrun). For SMOD=Match Store, indicates a match event occured before the previous match data was read from SHIFTBUF (SHIFTBUF Overrun).

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	For SMOD=Match Continuous, the error flag is set when a match has occured between SHIFTBUF and Shifter. Can be cleared by writing logic one to the flag. For SMOD=Match Continuous, can also be cleared when the SHIFTBUF register is read. 0000b - Shifter Error Flag is clear 0001b - Shifter Error Flag is set
4-31 Reserved	Reserved.

45.3.1.8 Timer Status Register (TIMSTAT)

45.3.1.8.1 Address

Register	Offset
TIMSTAT	18h

45.3.1.8.2 Function

.

45.3.1.8.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TSF				0											
W	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

45.3.1.8.4 Fields

Field	Function
0-3 TSF	Timer Status Flags

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	The timer status flag sets depending on the timer mode, and can be cleared by writing logic one to the flag. In 8-bit counter mode, the timer status flag is set when the upper 8-bit counter equals zero and decrements, this also causes the counter to reload with the value in the compare register. In 8-bit PWM mode, the timer status flag is set when the upper 8-bit counter equals zero and decrements, this also causes the counter to reload with the value in the compare register.. In 16-bit counter mode, the timer status flag is set when the 16-bit counter equals zero and decrements, this also causes the counter to reload with the value in the compare register.. 0000b - Timer Status Flag is clear 0001b - Timer Status Flag is set
4-31 Reserved	Reserved.

45.3.1.9 Shifter Status Interrupt Enable (SHIFTSIEN)

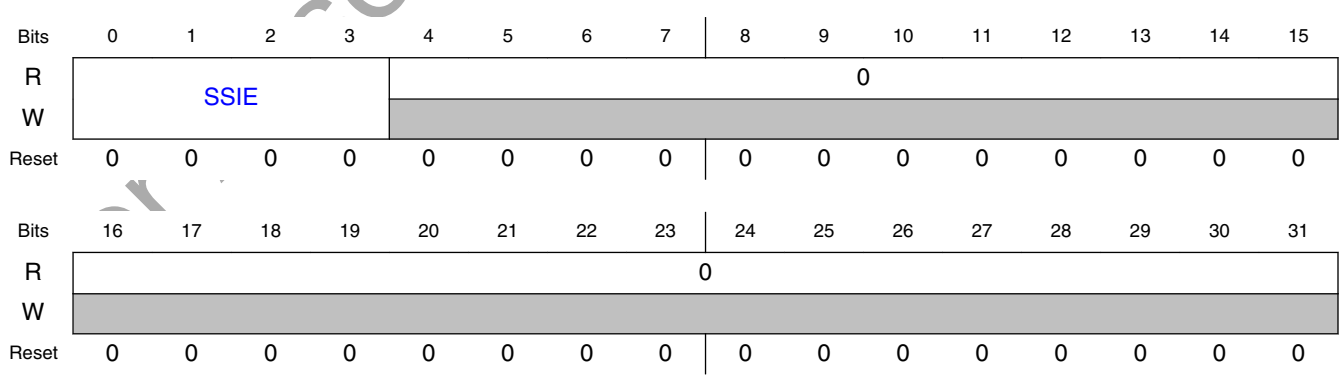
45.3.1.9.1 Address

Register	Offset
SHIFTSIEN	20h

45.3.1.9.2 Function

.

45.3.1.9.3 Diagram



Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.9.4 Fields

Field	Function
0-3 SSIE	Shifter Status Interrupt Enable Enables interrupt generation when corresponding SSF is set. 0000b - Shifter Status Flag interrupt disabled 0001b - Shifter Status Flag interrupt enabled
4-31 Reserved	Reserved.

45.3.1.10 Shifter Error Interrupt Enable (SHIFTEIEN)

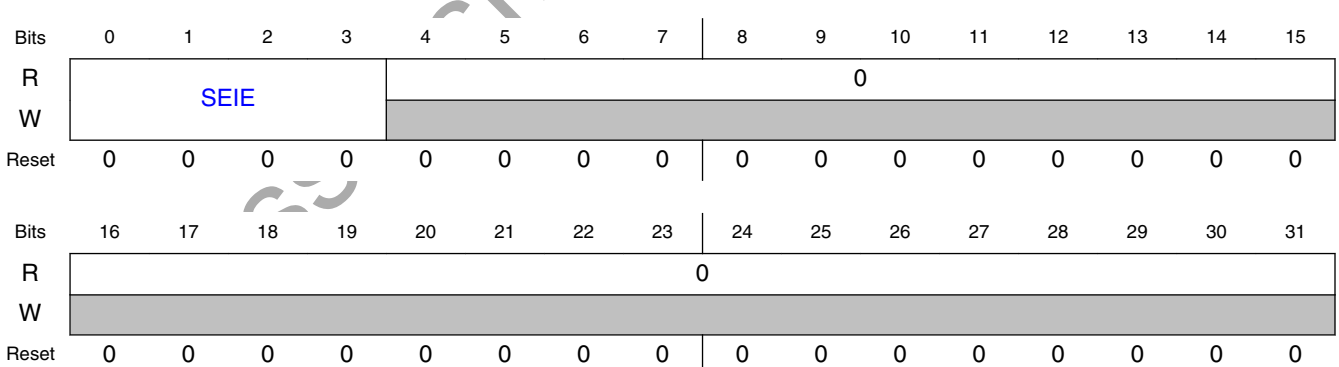
45.3.1.10.1 Address

Register	Offset
SHIFTEIEN	24h

45.3.1.10.2 Function

.

45.3.1.10.3 Diagram



45.3.1.10.4 Fields

Field	Function
0-3 SEIE	Shifter Error Interrupt Enable Enables interrupt generation when corresponding SEF is set.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0000b - Shifter Error Flag interrupt disabled 0001b - Shifter Error Flag interrupt enabled
4-31 Reserved	Reserved.

45.3.1.11 Timer Interrupt Enable Register (TIMIEN)

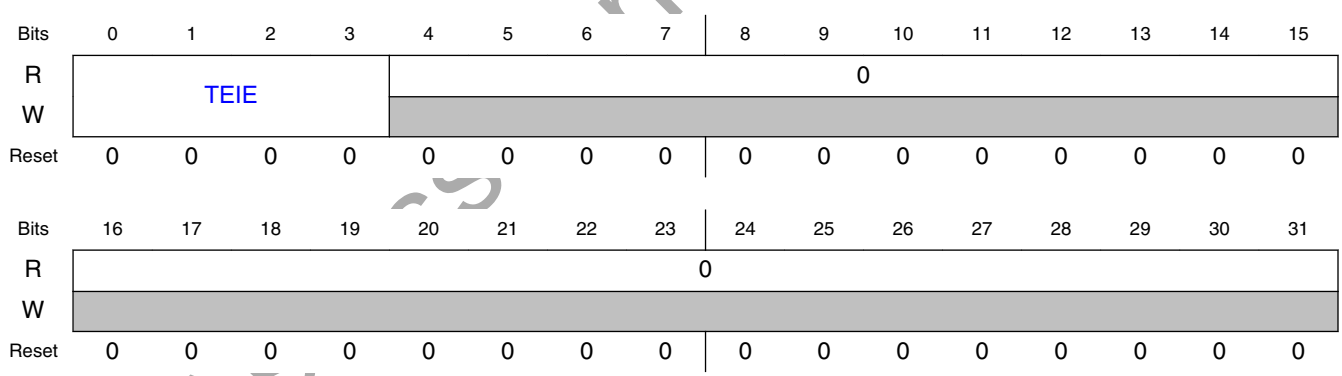
45.3.1.11.1 Address

Register	Offset
TIMIEN	28h

45.3.1.11.2 Function

.

45.3.1.11.3 Diagram



45.3.1.11.4 Fields

Field	Function
0-3 TEIE	Timer Status Interrupt Enable Enables interrupt generation when corresponding TSF is set. 0000b - Timer Status Flag interrupt is disabled 0001b - Timer Status Flag interrupt is enabled
4-31 Reserved	Reserved.

For Assignment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.12 Shifter Status DMA Enable (SHIFTSDEN)

45.3.1.12.1 Address

Register	Offset
SHIFTSDEN	30h

45.3.1.12.2 Function

.

45.3.1.12.3 Diagram

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

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

45.3.1.12.4 Fields

Field	Function
0-3 SSDE	Shifter Status DMA Enable Enables DMA request generation when corresponding SSF is set. 0000b - Shifter Status Flag DMA request is disabled 0001b - Shifter Status Flag DMA request is enabled
4-31 Reserved	Reserved.

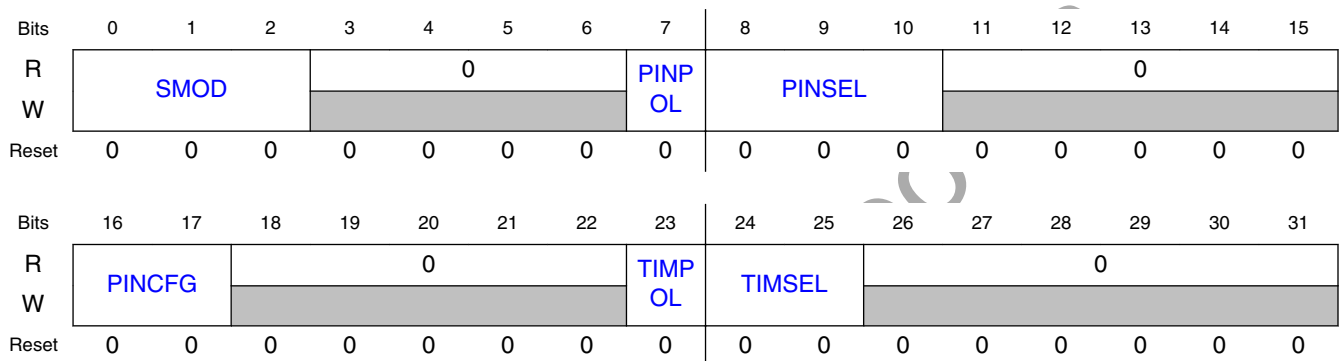
45.3.1.13 Shifter Control N (SHIFTCTLa)

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.13.1 Address

Register	Offset
SHIFTCTL0	80h
SHIFTCTL1	84h
SHIFTCTL2	88h
SHIFTCTL3	8Ch

45.3.1.13.2 Diagram



45.3.1.13.3 Fields

Field	Function
0-2 SMOD	Shifter Mode Configures the mode of the Shifter. 000b - Disabled. 001b - Receive mode. Captures the current Shifter content into the SHIFTBUF on expiration of the Timer. 010b - Transmit mode. Load SHIFTBUF contents into the Shifter on expiration of the Timer. 011b - Reserved. 100b - Match Store mode. Shifter data is compared to SHIFTBUF content on expiration of the Timer. 101b - Match Continuous mode. Shifter data is continuously compared to SHIFTBUF contents. 110b - 111b -
3-6 —	Reserved.
7 PINPOL	Shifter Pin Polarity 0b - Pin is active high 1b - Pin is active low
8-10 PINSEL	Shifter Pin Select Selects which pin is used by the Shifter input or output. PINSEL=i will select the FXIO_Di pin.
11-15	Reserved.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
—	
16-17 PINCFG	Shifter Pin Configuration 00b - Shifter pin output disabled 01b - Shifter pin open drain or bidirectional output enable 10b - Shifter pin bidirectional output data 11b - Shifter pin output
18-22 —	Reserved.
23 TIMPOL	Timer Polarity 0b - Shift on posedge of Shift clock 1b - Shift on negedge of Shift clock
24-25 TIMSEL	Timer Select Selects which Timer is used for controlling the logic/shift register and generating the Shift clock. TIMSEL=i will select TIMERi.
26-31 —	Reserved.

45.3.1.14 Shifter Configuration N (SHIFTCFGa)

45.3.1.14.1 Address

Register	Offset
SHIFTCFG0	100h
SHIFTCFG1	104h
SHIFTCFG2	108h
SHIFTCFG3	10Ch

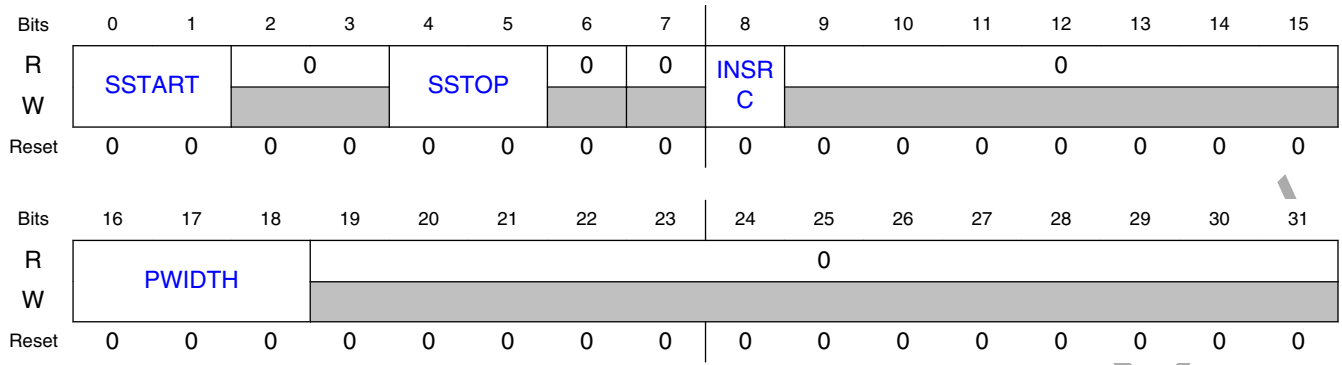
45.3.1.14.2 Function

.

For Assessment Purpose

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.14.3 Diagram



45.3.1.14.4 Fields

Field	Function
0-1 SSTART	Shifter Start bit For SMOD=Transmit, this field allows automatic start bit insertion if the selected timer has also enabled a start bit. For SMOD=Receive or Match Store, this field allows automatic start bit checking if the selected timer has also enabled a start bit. 00b - Start bit disabled for transmitter/receiver/match store, transmitter loads data on enable 01b - Start bit disabled for transmitter/receiver/match store, transmitter loads data on first shift 10b - Transmitter outputs start bit value 0 before loading data on first shift, receiver/match store sets error flag if start bit is not 0 11b - Transmitter outputs start bit value 1 before loading data on first shift, receiver/match store sets error flag if start bit is not 1
2-3 —	Reserved.
4-5 SSTOP	Shifter Stop bit For SMOD=Transmit, this field allows automatic stop bit insertion if the selected timer has also enabled a stop bit. For SMOD=Receive or Match Store, this field allows automatic stop bit checking if the selected timer has also enabled a stop bit. 00b - Stop bit disabled for transmitter/receiver/match store 01b - Reserved for transmitter/receiver/match store 10b - Transmitter outputs stop bit value 0 on store, receiver/match store sets error flag if stop bit is not 0 11b - Transmitter outputs stop bit value 1 on store, receiver/match store sets error flag if stop bit is not 1
6 —	Reserved.
7 —	Reserved.
8 INSRC	Input Source Selects the input source for the shifter.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	0b - Pin 1b - Shifter N+1 Output
9-15 —	Reserved.
16-18 PWIDTH	Parallel Width For all Shifters, this register field configures the number of bits to be shifted on each Shift clock as follows: 1-bit shift for PWIDTH=0 4-bit shift for PWIDTH=1...3 8-bit shift for PWIDTH=4...7 16-bit shift for PWIDTH=8...15 32-bit shift for PWIDTH=16...31 For Shifters which support parallel transmit (SHIFTER0, SHIFTER4) or parallel receive (SHIFTER3, SHIFTER7), this register field, together with SHIFTCTL[PINSEL], also selects the pins to be driven or sampled on each Shift clock as follows: FXIO_D[PINSEL+PWIDTH]:FXIO_D[PINSEL]
19-31 —	Reserved.

45.3.1.15 Shifter Buffer N (SHIFTBUFa)

45.3.1.15.1 Address

Register	Offset
SHIFTBUF0	200h
SHIFTBUF1	204h
SHIFTBUF2	208h
SHIFTBUF3	20Ch

45.3.1.15.2 Function

.

For Assessment Purpose

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.15.3 Diagram

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

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

45.3.1.15.4 Fields

Field	Function
0-31 SHIFTBUF	Shift Buffer Shift buffer data is used for a variety of functions depending on the SMOD setting: For SMOD=Receive, Shifter data is transferred into SHIFTBUF at the expiration of Timer. For SMOD=Transmit, SHIFTBUF data is transferred into the Shifter before the Timer begins. For SMOD=Match Store/Continuous, SHIFTBUF[31:16] contains the data to be matched with the Shifter contents. The Match is checked either continuosly (Match Continous mode) or when the Timer expires (Match Store mode). SHIFTBUF[15:0] can be used to mask the match result (1=mask, 0=no mask). In Match Store mode, Shifter data [31:16] is written to SHIFTBUF[31:16] whenever a match event occurs.

45.3.1.16 Shifter Buffer N Bit Swapped (SHIFTBUFBISa)

45.3.1.16.1 Address

Register	Offset
SHIFTBUFBIS0	280h
SHIFTBUFBIS1	284h
SHIFTBUFBIS2	288h
SHIFTBUFBIS3	28Ch

45.3.1.16.2 Function

.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.16.3 Diagram

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

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

45.3.1.16.4 Fields

Field	Function
0-31 SHIFTBUFBIS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register are bit swapped. Reads return SHIFTBUF[0:31].

45.3.1.17 Shifter Buffer N Byte Swapped (SHIFTBUFBYSa)

45.3.1.17.1 Address

Register	Offset
SHIFTBUFBYS0	300h
SHIFTBUFBYS1	304h
SHIFTBUFBYS2	308h
SHIFTBUFBYS3	30Ch

45.3.1.17.2 Function

.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.17.3 Diagram

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

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

45.3.1.17.4 Fields

Field	Function
0-31 SHIFTBUFBYS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register are byte swapped. Reads return { SHIFTBUF[7:0], SHIFTBUF[15:8], SHIFTBUF[23:16], SHIFTBUF[31:24] }.

45.3.1.18 Shifter Buffer N Bit Byte Swapped (SHIFTBUFBBSa)

45.3.1.18.1 Address

Register	Offset
SHIFTBUFBBS0	380h
SHIFTBUFBBS1	384h
SHIFTBUFBBS2	388h
SHIFTBUFBBS3	38Ch

45.3.1.18.2 Function

.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.18.3 Diagram

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

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

45.3.1.18.4 Fields

Field	Function
0-31 SHIFTBUFBBS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register are bit swapped within each byte. Reads return { SHIFTBUF[24:31], SHIFTBUF[16:23], SHIFTBUF[8:15], SHIFTBUF[0:7] }.

45.3.1.19 Timer Control N (TIMCTLa)

45.3.1.19.1 Address

Register	Offset
TIMCTL0	400h
TIMCTL1	404h
TIMCTL2	408h
TIMCTL3	40Ch

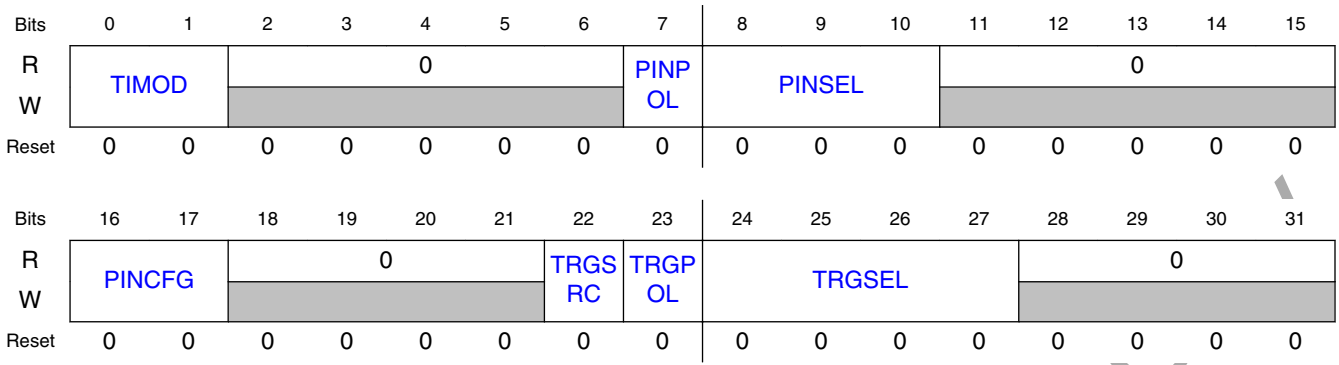
45.3.1.19.2 Function

.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.19.3 Diagram



45.3.1.19.4 Fields

Field	Function
0-1 TIMOD	Timer Mode In 8-bit counter mode, the lower 8-bits of the counter and compare register are used to configure the baud rate of the timer shift clock and the upper 8-bits are used to configure the shifter bit count. In 8-bit PWM mode, the lower 8-bits of the counter and compare register are used to configure the high period of the timer shift clock and the upper 8-bits are used to configure the low period of the timer shift clock. The shifter bit count is configured using another timer or external signal. In 16-bit counter mode, the full 16-bits of the counter and compare register are used to configure either the baud rate of the shift clock or the shifter bit count. 00b - Timer Disabled. 01b - Dual 8-bit counters baud/bit mode. 10b - Dual 8-bit counters PWM mode. 11b - Single 16-bit counter mode.
2-6 —	Reserved.
7 PINPOL	Timer Pin Polarity 0b - Pin is active high 1b - Pin is active low
8-10 PINSEL	Timer Pin Select Selects which pin is used by the Timer input or output. PINSEL=i will select the FXIO_Di pin.
11-15 —	Reserved.
16-17 PINCFG	Timer Pin Configuration 00b - Timer pin output disabled 01b - Timer pin open drain or bidirectional output enable 10b - Timer pin bidirectional output data 11b - Timer pin output
18-21 —	Reserved.
22	Trigger Source 0b - External trigger selected

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
TRGSRC	1b - Internal trigger selected
23 TRGPOL	Trigger Polarity 0b - Trigger active high 1b - Trigger active low
24-27 TRGSEL	Trigger Select The valid values for TRGSEL will depend on the FLEXIO_PARAM register. • When TRGSRC = 1, the valid values for N will depend on PIN, TIMER, SHIFTER fields in the FLEXIO_PARAM register. • When TRGSRC = 0, the valid values for N will depend on TRIGGER field in FLEXIO_PARAM register. Refer to the chip configuration section for external trigger selection. **NOTE:** For a pin, N=0 to 7. For a Shifter/Timer, N=0 to 3. The internal trigger selection is configured as follows: • 4*N - Pin 2*N input • 4*N+1 - Shifter N status flag • 4*N+2 - Pin 2*N+1 input • 4*N+3 - Timer N trigger output
28-31 —	Reserved.

45.3.1.20 Timer Configuration N (TIMCFGa)

45.3.1.20.1 Address

Register	Offset
TIMCFG0	480h
TIMCFG1	484h
TIMCFG2	488h
TIMCFG3	48Ch

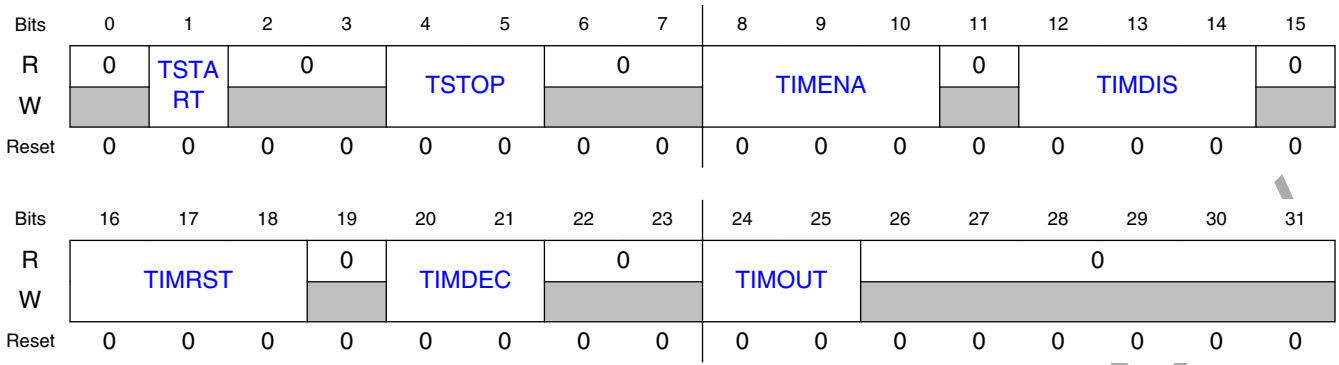
45.3.1.20.2 Function

The options to enable or disable the timer using the Timer N-1 enable or disable are reserved when N is evenly divisible by 4 (eg: Timer 0).

For Assessment Purpose

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.20.3 Diagram



45.3.1.20.4 Fields

Field	Function
0 —	Reserved.
1 TSTART	Timer Start Bit When start bit is enabled, configured shifters will output the contents of the start bit when the timer is enabled and the timer counter will reload from the compare register on the first rising edge of the shift clock. 0b - Start bit disabled 1b - Start bit enabled
2-3 —	Reserved.
4-5 TSTOP	Timer Stop Bit The stop bit can be added on a timer compare (between each word) or on a timer disable. When stop bit is enabled, configured shifters will output the contents of the stop bit when the timer is disabled. When stop bit is enabled on timer disable, the timer remains disabled until the next rising edge of the shift clock. If configured for both timer compare and timer disable, only one stop bit is inserted on timer disable. 00b - Stop bit disabled 01b - Stop bit is enabled on timer compare 10b - Stop bit is enabled on timer disable 11b - Stop bit is enabled on timer compare and timer disable
6-7 —	Reserved.
8-10 TIMENA	Timer Enable Configures the condition that causes the Timer to be enabled and start decrementing. 000b - Timer always enabled 001b - Timer enabled on Timer N-1 enable 010b - Timer enabled on Trigger high 011b - Timer enabled on Trigger high and Pin high 100b - Timer enabled on Pin rising edge 101b - Timer enabled on Pin rising edge and Trigger high 110b - Timer enabled on Trigger rising edge 111b - Timer enabled on Trigger rising or falling edge

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
11 —	Reserved.
12-14 TIMDIS	Timer Disable Configures the condition that causes the Timer to be disabled and stop decrementing. 000b - Timer never disabled 001b - Timer disabled on Timer N-1 disable 010b - Timer disabled on Timer compare 011b - Timer disabled on Timer compare and Trigger Low 100b - Timer disabled on Pin rising or falling edge 101b - Timer disabled on Pin rising or falling edge provided Trigger is high 110b - Timer disabled on Trigger falling edge 111b - Reserved
15 —	Reserved.
16-18 TIMRST	Timer Reset Configures the condition that causes the timer counter (and optionally the timer output) to be reset. In 8-bit counter mode, the timer reset will only reset the lower 8-bits that configure the baud rate. In all other modes, the timer reset will reset the full 16-bits of the counter. 000b - Timer never reset 001b - Reserved 010b - Timer reset on Timer Pin equal to Timer Output 011b - Timer reset on Timer Trigger equal to Timer Output 100b - Timer reset on Timer Pin rising edge 101b - Reserved 110b - Timer reset on Trigger rising edge 111b - Timer reset on Trigger rising or falling edge
19 —	Reserved.
20-21 TIMDEC	Timer Decrement Configures the source of the Timer decrement and the source of the Shift clock. 00b - Decrement counter on FlexIO clock, Shift clock equals Timer output. 01b - Decrement counter on Trigger input (both edges), Shift clock equals Timer output. 10b - Decrement counter on Pin input (both edges), Shift clock equals Pin input. 11b - Decrement counter on Trigger input (both edges), Shift clock equals Trigger input.
22-23 —	Reserved.
24-25 TIMOUT	Timer Output Configures the initial state of the Timer Output and whether it is affected by the Timer reset. 00b - Timer output is logic one when enabled and is not affected by timer reset 01b - Timer output is logic zero when enabled and is not affected by timer reset 10b - Timer output is logic one when enabled and on timer reset 11b - Timer output is logic zero when enabled and on timer reset
26-31 —	Reserved.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.3.1.21 Timer Compare N (TIMCMPa)

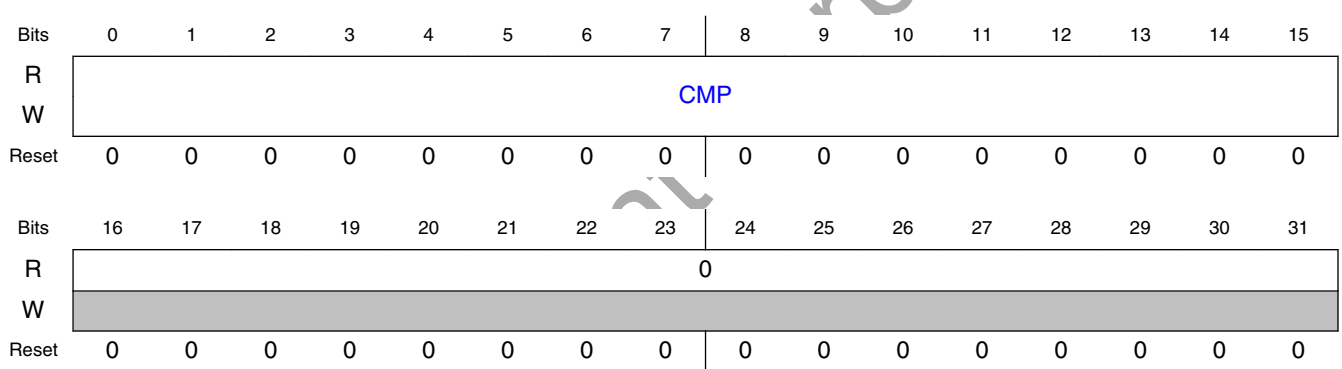
45.3.1.21.1 Address

Register	Offset
TIMCMP0	500h
TIMCMP1	504h
TIMCMP2	508h
TIMCMP3	50Ch

45.3.1.21.2 Function

.

45.3.1.21.3 Diagram



45.3.1.21.4 Fields

Field	Function
0-15 CMP	Timer Compare Value The timer compare value is loaded into the timer counter when the timer is first enabled, when the timer is reset and when the timer decrements down to zero. In dual 8-bit counters baud/bit mode, the lower 8-bits configures the baud rate divider equal to (CMP[7:0] + 1) * 2. The upper 8-bits configure the number of bits in each word equal to (CMP[15:8] + 1) / 2. In dual 8-bit counters PWM mode, the lower 8-bits configure the high period of the output to (CMP[7:0] + 1) and the upper 8-bits configure the low period of the output to (CMP[15:8] + 1). In 16-bit counter mode, the compare value can be used to generate the baud rate divider (if shift clock source is timer output) to equal (CMP[15:0] + 1) * 2. When the shift clock source is a pin or trigger input, the compare register is used to set the number of bits in each word equal to (CMP[15:0] + 1) / 2.
16-31 —	Reserved.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.4 Functional description

45.4.1 Shifter operation

Shifters are responsible for buffering and shifting data into or out of the FlexIO. The timing of shift, load and store events are controlled by the Timer assigned to the Shifter via the SHIFTCTL[TIMSEL] register. The Shifters are designed to support either DMA, interrupt or polled operation. The following block diagram provides a detailed view of the Shifter microarchitecture.

45.4.1.1 Transmit Mode

When configured for Transmit mode (SHIFTCTL[SMOD]=Transmit), the shifter will load data from the SHIFTBUF register and shift data out when a load event is signalled by the assigned Timer. An optional start/stop bit can also be automatically loaded before/after SHIFTBUF data by configuring the SHIFTCFG[SSTART], TIMCFG[TSTART] or SHIFTCFG[SSTOP], TIMCFG[TSTOP] registers in the Shifter and Timer. Note that the shifter will immediately load a stop bit when the Shifter is initially configured for Transmit mode if a stop bit is enabled.

The Shifter Status Flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests will set when data has been loaded from the SHIFTBUF register into the Shifter or when the Shifter is initially configured into Transmit mode. The flag will clear when new data has been written into the SHIFTBUF register.

The Shifter Error Flag (SHIFTERR[SEF]) and any enabled interrupts will set when an attempt to load data from an empty SHIFTBUF register occurs (buffer underrun). The flag can be cleared by writing it with logic 1.

45.4.1.2 Receive Mode

When configured for Receive mode (SHIFTCTL[SMOD]=Receive), the shifter will shift data in and store data into the SHIFTBUF register when a store event is signalled by the assigned Timer. Checking for a start/stop bit can be enabled before/after shifter data is sampled by configuring the SHIFTCFG[SSTART], TIMCFG[TSTART] or SHIFTCFG[SSTOP], TIMCFG[TSTOP] registers in the Shifter and Timer.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The Shifter Status Flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests will set when data has been stored into the SHIFTBUF register from the Shifter. The flag will clear when the data has been read from the SHIFTBUF register.

The Shifter Error Flag (SHIFTERR[SEF]) and any enabled interrupts will set when an attempt to store data into a full SHIFTBUF register occurs (buffer overrun) or when a mismatch occurs on a start/stop bit check. The flag can be cleared by writing it with logic 1.

45.4.1.3 Match Store Mode

When configured for Match Store mode (SHIFTCTL[SMOD]=Match Store), the shifter will shift data in, check for a match result and store matched data into the SHIFTBUF register when a store event is signalled by the assigned Timer. Checking for a start/stop bit can be enabled before/after shifter data is sampled by configuring the SHIFTCFG[SSTART], TIMCFG[TSTART] or SHIFTCFG[SSTOP], TIMCFG[TSTOP] registers in the Shifter and Timer. Up to 16-bits of data can be compared using SHIFTBUF[31:16] to configure the data to be matched and SHIFTBUF[15:0] to mask the match result.

The Shifter Status Flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests will set when a match occurs and matched data has been stored into the SHIFTBUF register from the Shifter. The flag will clear when the matched data has been read from the SHIFTBUF register.

The Shifter Error Flag (SHIFTERR[SEF]) and any enabled interrupts will set when an attempt to store matched data into a full SHIFTBUF register occurs (buffer overrun) or when a mismatch occurs on a start/stop bit check. The flag can be cleared by writing it with logic 1.

45.4.1.4 Match Continuous Mode

When configured for Match Continuous mode (SHIFTCTL[SMOD]=Match Continuous), the shifter will shift data in and continuously check for a match result whenever a shift event is signalled by the assigned Timer. Up to 16-bits of data can be compared using SHIFTBUF[31:16] to configure the data to be matched and SHIFTBUF[15:0] to mask the match result.

The Shifter Status Flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests will set when a match occurs. The flag will clear automatically as soon as there is no longer a match between Shifter data and SHIFTBUF register.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The Shifter Error Flag (SHIFTERR[SEF]) and any enabled interrupts will set when a match occurs. The flag will clear when there is a read from the SHIFTBUF register or it written with logic 1.

45.4.2 Timer operation

The FlexIO 16-bit timers control the loading, shifting and storing of the shift registers, the counters load the contents of the compare register and decrement down to zero on the FlexIO clock. They can perform generic timer functions such as generating a clock or select output or a PWM waveform. Timers can be configured to enable in response to a trigger, pin or shifter condition; decrement always or only on a trigger or pin edge; reset in response to a trigger or pin condition; and disable on a trigger or pin condition or on a timer compare. Timers can optionally include a start condition and/or stop condition.

Each timer operates independently, although a timer can be configured to enable or disable at the same time as the previous timer (eg: timer1 can enable or disable at the same time as timer 0) and a timer output can be used to trigger any other timer. The trigger used by each timer is configured independently and can be configured to be a timer output, shifter status flag, pin input or an external trigger input (refer to the chip configuration section for details on the external trigger connections). The trigger configuration is separate from the pin configuration, which can be configured for input, output data or output enable.

The Timer Configuration Register (TIMCFGn) should be configured before setting the Timer Mode (TIMOD). Once the TIMOD is configured for the desired mode, when the condition configured by timer enable (TIMENA) is detected then the following events occur.

- Timer counter will load the current value of the Compare Register and start decrementing as configured by TIMDEC.
- Timer output will set depending on the TIMOUT configuration.
- Transmit shifters controlled by this timer will either output their start bit value, or load the shift register from the shift buffer and output the first bit, as configured by SSTART.

The Timer will then generate the timer output and timer shift clock depending on the TIMOD and TIMDEC fields. The shifter clock is either equal to the timer output (when TIMDEC=00 or 01) or equal to the decrement clock (when TIMDEC=10 or 11). When TIMDEC is configured to decrement from a pin or trigger, the timer will decrement on both rising and falling edges.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

When the Timer is configured to reset as configured in the TIMRST field then the Timer counter will load the current value of the Compare Register again, the timer output may also be affected by the reset as configured in TIMOUT.

If the Timer start bit is enabled, the timer counter will reload with the compare register on the first rising edge of the shift clock after the timer starts decrementing. If there is no falling edge on the shift clock before the first rising edge (for example, when TIMOUT=1), a shifter that is configured to shift on falling edge and load on the first shift will not load correctly.

When configured for 8-bit counter mode, whenever the lower 8-bit counter decrements to zero the timer output will toggle, the lower 8-bit counter register will reload from the compare register and the upper 8-bit counter will decrement. For 8-bit PWM mode, the lower 8-bit counter will only decrement when the output is high and the upper 8-bit counter will only decrement when the output is low. The timer output will toggle whenever either lower or upper 8-bit counter decrements to zero.

When the timer decrements to zero, a compare event occurs depending on the timer mode. For 8-bit counter or PWM modes, both halves of the counter must equal zero and the upper half must decrement for the timer compare event to occur, while in 16-bit mode the entire counter must equal zero and decrement. The timer compare event will cause the timer status flag to set, the timer counter to load the contents of the timer compare register, the timer output to toggle, any configured transmit shift registers to load and any configured receive shift registers to store.

When the is Timer is configured to add a stop bit on each compare, the following additional events will occur.

- Transmit shifters controlled by this timer will output their stop bit value (if configured by SSTOP).
- Receive shifters controlled by this timer will store the contents of the shift register in their shift buffer, as configured by SSTOP.
- On the first rising edge of the shifter clock after the compare, the timer counter will reload the current value of the Compare Register.

Transmit shifters must be configured to load on the first shift when the timer is configured to insert a stop bit on each compare.

When the condition configured by timer disable (TIMDIS) is detected, the following events occur.

- Timer counter will reload the current value of the Compare Register and start decrementing as configured by TIMDEC.
- Timer output will clear.

For Assessment Purposes Only

- Transmit shifters controlled by this timer will output their stop bit value (if configured by SSTOP).
- Receive shifters controlled by this timer will store the contents of the shift register in their shift buffer, as configured by SSTOP.

If the timer stop bit is enabled, the timer counter will continue decrementing until the next rising edge of the shift clock is detected, at which point it will finish. A timer enable condition can be detected in the same cycle as a timer disable condition (if timer stop bit is disabled), or on the first rising edge of the shift clock after the disable condition (if stop bit is enabled). Receive shift registers will stop bit enabled will store the contents of the shift register into the shift buffer and verify the state of the input data on the configured shift edge while the timer is in the stop state condition. If there is no configured edge between the timer disable and the next rising edge of the shift clock then the final store and verify do not occur.

45.4.3 Pin operation

The pin configuration for each timer and shifter can be configured to use any FlexIO pin with either polarity. Each timer and shifter can be configured as an input, output data, output enable or bidirectional output. A pin configured for output enable can be used as an open drain (with inverted polarity, since the output enable assertion would cause logic zero to be output on the pin) or to control the enable on the bidirectional output. Any timer or shifter could be configured to control the output enable for a pin where the bidirectional output data is driven by another timer or shifter.

When configuring a pin as an input (this includes a timer trigger configured as a pin input), the input signal is first synchronized to the FlexIO clock before the signal is used by a timer or shifter. This introduces a small latency of between 0.5 to 1.5 FlexIO clock cycles when using an external pin input to generate an output or control a shifter. This sets the maximum setup time at 1.5 FlexIO clock cycles.

If an input is used by more than one timer or shifter then the synchronization occurs once to ensure any edge is seen on the same cycle by all timers and shifters using that input.

Note that FlexIO pins are also connected internally, configuring a FlexIO shifter or timer to output data on an unused pin will make an internal connection that allows other shifters and timer to use this pin as an input. This allows a shifter output to be used to trigger a timer or a timer output to be shifted into a shifter. This path is also synchronized to the FlexIO clock and therefore incurs a 1 cycle latency.

So when using a Pin input as a Timer Trigger, Timer Clock or Shifter Data Input, the following synchronization delays occur:

1. 0.5 – 1.5 FlexIO clock cycles for external pin

For Assessment Purposes Only

2. 1 FlexIO clock cycle for an internally driven pin

For timing considerations such as output valid time and input setup time for specific applications (SPI Master, SPI Slave, I2C Master, I2S Master, I2S Slave) please refer to the FlexIO Application Information Section.

45.5 Application Information

This section provides examples for a variety of FlexIO module applications.

45.5.1 UART Transmit

UART transmit can be supported using one Timer, one Shifter and one Pin (two Pins if supporting CTS). The start and stop bit insertion is handled automatically and multiple transfers can be supported using DMA controller. The timer status flag can be used to indicate when the stop bit of each word is transmitted.

Break and idle characters require software intervention, before transmitting a break or idle character the SSTART and SSTOP fields should be altered to transmit the required state and the data to transmit must equal 0xFF or 0x00. Supporting a second stop bit requires the stop bit to be inserted into the data stream using software (and increasing the number of bits to transmit). Note that when performing byte writes to SHIFTBUFn (or SHIFTBUFBIS for transmitting MSB first), the rest of the register remains unaltered allowing an address mark bit or additional stop bit to remain undisturbed.

FlexIO does not support automatic insertion of parity bits.

Table 45-3. UART Transmit Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0032	Configure start bit of 0 and stop bit of 1.
SHIFTCTLn	0x0003_0002	Configure transmit using Timer 0 on posedge of clock with output data on Pin 0. Can invert output data by setting PINPOL, or can support open drain by setting PINPOL=0x1 and PINCFG=0x1.
TIMCMPn	0x0000_0F01	Configure 8-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0000_2222	Configure start bit, stop bit, enable on trigger low and disable on compare. Can support CTS by configuring TIMEN=0x3.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required
Review Draft
Freescale Semiconductor, Inc.
Assignment Purposes Only

Table 45-3. UART Transmit Configuration (continued)

Register	Value	Comments
TIMCTLn	0x01C0_0001	Configure dual 8-bit counter using Shifter 0 status flag as inverted internal trigger source. Can support CTS by configuring PINSEL=0x1 (for Pin 1) and PINPOL=0x1.
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF[7:0] to initiate an 8-bit transfer, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS[7:0] register instead.

45.5.2 UART Receive

UART receive can be supported using one Timer, one Shifter and one Pin (two Timers and two Pins if supporting RTS). The start and stop bit verification is handled automatically and multiple transfers can be supported using the DMA controller. The timer status flag can be used to indicate when the stop bit of each word is received.

Triple voting of the received data is not supported by FlexIO, data is sampled only once in the middle of each bit. Another timer can be used to implement a glitch filter on the incoming data, another Timer can also be used to detect an idle line of programmable length. Break characters will cause the error flag to set and the shifter buffer register will return 0x00.

FlexIO does not support automatic verification of parity bits.

ssment Purposes

Table 45-4. UART Receiver Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0032	Configure start bit of 0 and stop bit of 1.
SHIFTCTLn	0x0080_0001	Configure receive using Timer 0 on negedge of clock with input data on Pin 0. Can invert input data by setting PINPOL.
TIMCMPn	0x0000_0F01	Configure 8-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0204_2422	Configure start bit, stop bit, enable on pin posedge and disable on compare. Enable resynchronization to received

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 45-4. UART Receiver Configuration (continued)

Register	Value	Comments
		data with TIMOUT=0x2 and TIMRST=0x4.
TIMCTLn	0x0000_0081	Configure dual 8-bit counter using inverted Pin 0 input.
SHIFTBUFn	Data to receive	Received data can be read from SHIFTBUFBYS[7:0], use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS[7:0] register instead.

The UART Receiver with RTS configuration uses a 2nd Timer to generate the RTS output. The RTS will assert when the start bit is detected and negate when the data is read from the shifter buffer register. No start bit will be detected while the RTS is asserted, the received data is simply ignored.

Table 45-5. UART Receiver with RTS Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0032	Configure start bit of 0 and stop bit of 1.
SHIFTCTLn	0x0080_0001	Configure receive using Timer 0 on negedge of clock with input data on Pin 0. Can invert input data by setting PINPOL.
TIMCMPn	0x0000_0F01	Configure 8-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0204_2522	Configure start bit, stop bit, enable on pin posedge with trigger low and disable on compare. Enable resynchronization to received data with TIMOUT=0x2 and TIMRST=0x4.
TIMCTLn	0x03C0_0081	Configure dual 8-bit counter using inverted Pin 0 input. Trigger is internal using inverted Pin 1 input.
TIMCMP(n+1)	0x0000_FFFF	Never compare.
TIMCFG(n+1)	0x0030_6100	Enable on Timer N enable and disable on trigger falling edge. Decrement on trigger to ensure no compare.
TIMCTL(n+1)	0x0143_0083	Configure 16-bit counter and output on Pin 1. Trigger is internal using Shifter 0 flag.
SHIFTBUFn	Data to receive	Received data can be read from SHIFTBUFBYS[7:0], use the Shifter Status Flag to indicate when data can be

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 45-5. UART Receiver with RTS Configuration

Register	Value	Comments
		read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS[7:0] register instead.

45.5.3 SPI Master

SPI master mode can be supported using two Timers, two Shifters and four Pins. Either CPHA=0 or CPHA=1 can be supported and transfers can be supported using the DMA controller. For CPHA=1, the select can remain asserted for multiple transfers and the timer status flag can be used to indicate the end of the transfer.

The stop bit is used to guarantee a minimum of 1 clock cycle between the slave select negating and before the next transfer. Writing to the transmit buffer by either core or DMA is used to initiate each transfer.

Due to synchronization delays, the setup time for the serial input data is 1.5 FlexIO clock cycles, so the maximum baud rate is divide by 4 of the FlexIO clock frequency.

+ Purposes Only

Table 45-6. SPI Master (CPHA=0) Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0000	Start and stop bit disabled.
SHIFTCTLn	0x0083_0002	Configure transmit using Timer 0 on negedge of clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0000_0101	Configure receive using Timer 0 on posedge of clock with input data on Pin 1.
TIMCMPn	0x0000_3F01	Configure 32-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0100_2222	Configure start bit, stop bit, enable on trigger high and disable on compare, initial clock state is logic 0. Set PINPOL to invert the output shift clock.
TIMCTLn	0x01C3_0201	Configure dual 8-bit counter using Pin 2 output (shift clock), with Shifter 0 flag as the inverted trigger.
TIMCMP(n+1)	0x0000_FFFF	Never compare.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 45-6. SPI Master (CPHA=0) Configuration (continued)

Register	Value	Comments
TIMCFG(n+1)	0x0000_1100	Enable when Timer 0 is enabled and disable when Timer 0 is disabled.
TIMCTL(n+1)	0x0003_0383	Configure 16-bit counter (never compare) using inverted Pin 3 output (as slave select).
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS register instead.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBYS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS register instead.

Table 45-7. SPI Master (CPHA=1) Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0021	Start bit loads data on first shift.
SHIFTCTLn	0x0003_0002	Configure transmit using Timer 0 on posedge of clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0080_0101	Configure receive using Timer 0 on negedge of clock with input data on Pin 1.
TIMCMPn	0x0000_3F01	Configure 32-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0100_2222	Configure start bit, stop bit, enable on trigger high and disable on compare, initial clock state is logic 0. Set PINPOL to invert the output shift clock. Set TIMDIS=3 to keep slave select asserted for as long as there is data in the transmit buffer.
TIMCTLn	0x01C3_0201	Configure dual 8-bit counter using Pin 2 output (shift clock), with Shifter 0 flag as the inverted trigger.
TIMCMP(n+1)	0x0000_FFFF	Never compare.
TIMCFG(n+1)	0x0000_1100	Enable when Timer 0 is enabled and disable when Timer 0 is disabled.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 45-7. SPI Master (CPHA=1) Configuration (continued)

Register	Value	Comments
TIMCTL(n+1)	0x0003_0383	Configure 16-bit counter (never compare) using inverted Pin 3 output (as slave select).
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS register instead.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBYS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS register instead.

45.5.4 SPI Slave

SPI slave mode can be supported using one Timer, two Shifters and four Pins. Either CPHA=0 or CPHA=1 can be supported and transfers can be supported using the DMA controller. For CPHA=1, the select can remain asserted for multiple transfers and the timer status flag can be used to indicate the end of the transfer.

The transmit data must be written to the transmit buffer register before the external slave select asserts, otherwise the shifter error flag will be set.

Due to synchronization delays, the output valid time for the serial output data is 2.5 FlexIO clock cycles, so the maximum baud rate is divide by 6 of the FlexIO clock frequency.

Assessment Purpos

Table 45-8. SPI Slave (CPHA=0) Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0000	Start and stop bit disabled.
SHIFTCTLn	0x0083_0002	Configure transmit using Timer 0 on falling edge of shift clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0000_0101	Configure receive using Timer 0 on rising edge of shift clock with input data on Pin 1.
TIMCMPn	0x0000_003F	Configure 32-bit transfer. Set TIMCMP[15:0] = (number of bits x 2) - 1.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 45-8. SPI Slave (CPHA=0) Configuration (continued)

Register	Value	Comments
TIMCFGn	0x0120_6000	Configure enable on trigger rising edge, initial clock state is logic 0 and decrement on pin input.
TIMCTLn	0x06C0_0203	Configure 16-bit counter using Pin 2 input (shift clock), with Pin 3 input (slave select) as the inverted trigger.
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS register instead.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBYS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS register instead.

Table 45-9. SPI Slave (CPHA=1) Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0001	Shifter configured to load on first shift and stop bit disabled.
SHIFTCTLn	0x0003_0002	Configure transmit using Timer 0 on rising edge of shift clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0080_0101	Configure receive using Timer 0 on falling edge of shift clock with input data on Pin 1.
TIMCMPn	0x0000_003F	Configure 32-bit transfer. Set TIMCMP[15:0] = (number of bits x 2) - 1.
TIMCFGn	0x0120_6602	Configure start bit, enable on trigger rising edge, disable on trigger falling edge, initial clock state is logic 0 and decrement on pin input.
TIMCTLn	0x06C0_0203	Configure 16-bit counter using Pin 2 input (shift clock), with Pin 3 input (slave select) as the inverted trigger.
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS register instead.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBYS, use the Shifter Status Flag to indicate when data can be read

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 45-9. SPI Slave (CPHA=1) Configuration

Register	Value	Comments
		using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS register instead.

45.5.5 I2C Master

I2C master mode can be supported using two Timers, two Shifters and two Pins. One timer is used to generate the SCL output and one timer is used to control the shifters. The two shifters are used to transmit and receive for every word, when receiving the transmitter must transmit 0xFF to tristate the output. FlexIO inserts a stop bit after every word to generate/verify the ACK/NACK. FlexIO waits for the first write to the transmit data buffer before enabling SCL generation. Data transfers can be supported using the DMA controller and the shifter error flag will set on transmit underrun or receive overflow.

The first timer generates the bit clock for the entire packet (START to Repeated START/ STOP), so the compare register needs to be programmed with the total number of clock edges in the packet (minus one). The timer supports clock stretching using the reset counter when pin equal to output (although this increases both the clock high and clock low periods by at least 1 FlexIO clock cycle each). The second timer uses the SCL input pin to control the transmit/receive shift registers, this enforces an SDA data hold time by an extra 2 FlexIO clock cycles.

Both the transmit and receive shifters need to be serviced for each word in the transfer, the transmit shifter must transmit 0xFF when receiving and the receive shifter returns the data actually present on the SDA pin. The transmit shifter will load 1 additional word on the last falling edge of SCL pin, this word should be 0x00 if generating a STOP condition or 0xFF if generating a repeated START condition. During the last word of a master-receiver transfer, the transmit SSTOP bit should be set by software to generate a NACK.

The receive shift register will assert an error interrupt if a NACK is detected, but software is responsible for generating the STOP or repeated START condition. If a NACK is detected during master-transmit, the interrupt routine should immediately write the transmit shifter register with 0x00 (if generating STOP) or 0xFF (if generating repeated START). Software should then wait for the next rising edge on SCL and then disable both timers. The transmit shifter should then be disabled after waiting the setup delay for a repeated START or STOP condition.

Due to synchronization delays, the data valid time for the transmit output is 2 FlexIO clock cycles, so the maximum baud rate is divide by 6 of the FlexIO clock frequency.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The I2C master data valid is delayed 2 cycles because the clock output is passed through a synchronizer before clocking the transmit/receive shifter (to guarantee some SDA hold time). Since the SCL output is synchronous with FlexIO clock, the synchronization delay is 1 cycle and then 1 cycle to generate the output.

Table 45-10. I2C Master Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0032	Start bit enabled (logic 0) and stop bit enabled (logic 1).
SHIFTCTLn	0x0101_0082	Configure transmit using Timer 1 on rising edge of clock with inverted output enable (open drain output) on Pin 0.
SHIFTCFG(n+1)	0x0000_0020	Start bit disabled and stop bit enabled (logic 0) for ACK/NACK detection.
SHIFTCTL(n+1)	0x0180_0001	Configure receive using Timer 1 on falling edge of clock with input data on Pin 0.
TIMCMPn	0x0000_2501	Configure 2 word transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of words x 18) + 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0102_2222	Configure start bit, stop bit, enable on trigger high, disable on compare, reset if output equals pin. Initial clock state is logic 0 and is not affected by reset.
TIMCTLn	0x01C1_0101	Configure dual 8-bit counter using Pin 1 output enable (SCL open drain), with Shifter 0 flag as the inverted trigger.
TIMCMP(n+1)	0x0000_000F	Configure 8-bit transfer. Set TIMCMP[15:0] = (number of bits x 2) - 1.
TIMCFG(n+1)	0x0020_1112	Enable when Timer 0 is enabled, disable when Timer 0 is disabled, enable start bit and stop bit at end of each word, decrement on pin input.
TIMCTL(n+1)	0x01C0_0183	Configure 16-bit counter using inverted Pin 1 input (SCL).
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUFBBS[7:0], use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBIS[7:0], use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

45.5.6 I2S Master

I2S master mode can be supported using two Timers, two Shifters and four Pins. One timer is used to generate the bit clock and control the shifters and one timer is used to generate the frame sync. FlexIO waits for the first write to the transmit data buffer before enabling bit clock and frame sync generation. Data transfers can be supported using the DMA controller and the shifter error flag will set on transmit underrun or receive overflow.

The bit clock frequency is an even integer divide of the FlexIO clock frequency, and the initial frame sync assertion occurs at the same time as the first bit clock edge. The timer uses the start bit to ensure the frame sync is generated one clock cycle before the first output data.

Due to synchronization delays, the setup time for the receiver input is 1.5 FlexIO clock cycles, so the maximum baud rate is divide by 4 of the FlexIO clock frequency.

oses Only

Table 45-11. I2S Master Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0001	Load transmit data on first shift and stop bit disabled.
SHIFTCTLn	0x0003_0002	Configure transmit using Timer 0 on rising edge of clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0080_0101	Configure receive using Timer 0 on falling edge of clock with input data on Pin 1.
TIMCMPn	0x0000_3F01	Configure 32-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0000_0202	Configure start bit, enable on trigger high and never disable. Initial clock state is logic 1.
TIMCTLn	0x01C3_0201	Configure dual 8-bit counter using Pin 2 output (bit clock), with Shifter 0 flag as the inverted trigger. Set PINPOL to invert the output shift clock.
TIMCMP(n+1)	0x0000_007F	Configure 32-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:0] = (number of bits x baud rate divider) - 1.
TIMCFG(n+1)	0x0000_0100	Enable when Timer 0 is enabled and never disable.
TIMCTL(n+1)	0x0003_0383	Configure 16-bit counter using inverted Pin 3 output (as frame sync).

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 45-11. I2S Master Configuration (continued)

Register	Value	Comments
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUFBIS, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support LSB first transfer by writing to SHIFTBUF register instead.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBIS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support LSB first transfer by reading from SHIFTBUF register instead.

45.5.7 I2S Slave

I2S slave mode can be supported using two Timers, two Shifters and four Pins (for single transmit and single receive, other combinations of transmit and receive are possible).

The transmit data must be written to the transmit buffer register before the external frame sync asserts, otherwise the shifter error flag will be set.

Due to synchronization delays, the output valid time for the serial output data is 2.5 FlexIO clock cycles, so the maximum baud rate is divide by 6 of the FlexIO clock frequency.

The output valid time of I2S slave is max 2.5 cycles because there is a maximum 1.5 cycle delay on the clock synchronization plus 1 cycle to output the data

ssment Purposes

Table 45-12. I2S Slave Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0000	Start and stop bit disabled.
SHIFTCTLn	0x0103_0002	Configure transmit using Timer 1 on rising edge of shift clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0180_0101	Configure receive using Timer 1 on falling edge of shift clock with input data on Pin 1.
TIMCMPn	0x0000_007D	Configure two 32-bit transfers per frame. Set TIMCMP[15:0] = (number of bits x 4) - 3.
TIMCFGn	0x0030_2400	Configure enable on pin rising edge (inverted frame sync) and disable on

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 45-12. I2S Slave Configuration (continued)

Register	Value	Comments
		compare, initial clock state is logic 1 and decrement on trigger input (bit clock).
TIMCTLn	0x0440_0383	Configure 16-bit counter using inverted Pin 3 input (frame sync), with Pin 2 input (bit clock) as the trigger.
TIMCMP(n+1)	0x0000_003F	Configure 32-bit transfers. Set TIMCMP[15:0] = (number of bits x 2) - 1.
TIMCFG(n+1)	0x0020_3500	Configure enable on pin rising edge with trigger high and disable on compare with trigger low, initial clock state is logic 0 and decrement on pin input.
TIMCTL(n+1)	0x0340_0203	Configure 16-bit counter using Pin 2 input (bit clock), with Timer 0 output as the trigger.
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUFBIS, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support LSB first transfer by writing to SHIFTBUF register instead.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBIS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support LSB first transfer by reading from SHIFTBUF register instead.

For Assessme

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 46
FlexCAN

46.1 Chip-specific FlexCAN information

46.1.1 FlexCAN message buffer memory map

The FlexCAN memory buffers are allocated in memory according to the tables below.

Table 46-1. 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...

poses Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-1. 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 46-2. 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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-3. 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 46-4. 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

For Asse

46.1.2 Reset value of MDIS bit

The CAN_MCR[MDIS] bit is set after reset. Therefore, FlexCAN module is disabled following a reset.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

46.1.3 Number of message buffers

This device contains 3 FlexCAN modules. The number of message buffers for each FlexCAN module are listed in following table. Each message buffer is 16 bytes.

Table 46-5. Number of message buffers

Module Names	Number of Message Buffers (MB)	ISO CAN FD feature
FlexCAN0	32 MBs	Yes
FlexCAN1	16 MBs	No
FlexCAN2	16 MBs	No

46.1.4 FlexCAN Interrupts

The FlexCAN has multiple sources of interrupt requests. However, some of these sources are OR'd together to generate a single interrupt request. See below for the mapping of the individual interrupt sources to the interrupt request:

Request	Sources
Message buffer	Message buffers 0-15
Bus off	Bus off
Error	• Bit1 error • Bit0 error • Acknowledge error • Cyclic redundancy check (CRC) error • Form error • Stuffing error • Transmit error warning • Receive error warning
Transmit Warning	Transmit Warning
Receive Warning	Receive Warning
Wake-up	Wake-up

46.1.5 FlexCAN Operation in Low Power Modes

The FlexCAN module is operational in VLPR and VLPW modes. With the 4 MHz bus clock (in VLPR mode), the fastest supported FlexCAN transfer rate is 250 kbps. The bit timing parameters in the module must be adjusted for the new frequency, but full functionality is possible.

For Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

The FlexCAN module can be configured to generate a wakeup interrupt in STOP and VLPS modes. When the FlexCAN is configured to generate a wakeup, a recessive to dominant transition on the CAN bus generates an interrupt.

46.1.6 FlexCAN Doze Mode

The Doze mode for the FlexCAN module is the same as the Wait and VLPW modes for the chip.

46.2 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 FIFOs cannot be used in FD mode.

For Assessment Purposes Only

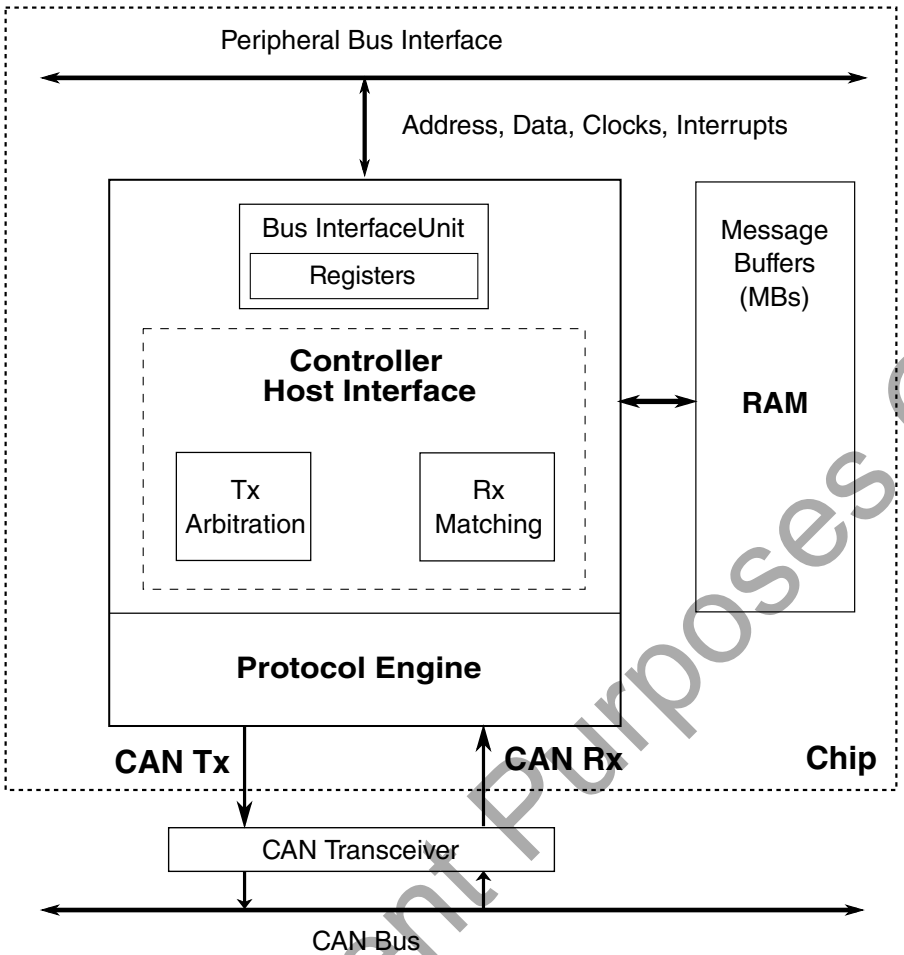


Figure 46-1. FlexCAN block diagram

46.2.1 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

46.2.2 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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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 or matching with received frames (Pretended Networking)
- 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

For Assessment Purposes Only

- 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
- Supports Pretended Networking functionality in low power: Stop mode

46.2.3 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

For Assessment Purposes Only

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.

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.

- 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.

- Pretended Network Mode:

  This mode can be selected to operate together with Doze or Stop Modes. Before entering in one of these low power modes the PNET_EN bit in MCR Register must be asserted. Once in low power mode, CHI sub-block clocks are shut down and CAN_PE sub-block is kept clocked, so that the Rx receive process is still active to filter incoming messages (see Receive Process under Pretended Networking Mode) as defined by the configuration registers (see Pretended Networking Control 1 Register (CTRL1_PN)). Upon detecting a wake up event, a Wake Up interrupt is issued to the system. When PNET_EN bit in MCR is asserted, the CPU must disable the Self Wake Up feature by negating SLF_WAK bit in MCR register (see Module Configuration Register (MCR)).

For Assessment Purposes Only

46.3 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 46-6. FlexCAN signal descriptions

Signal	Description	I/O
CAN Rx	CAN Receive Pin	Input
CAN Tx	CAN Transmit Pin	Output

46.3.1 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.

46.3.2 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.

46.4 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.

46.4.1 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 46-7.

Table 46-7. 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
Pretended Networking Control 1 register (CAN_CTRL1_PN)	S/U	Yes	Yes
Pretended Networking Control 2 register (CAN_CTRL2_PN)	S/U	Yes	Yes
Pretended Networking Wake Up Match register (CAN_WU_MTC)	S/U	Yes	Yes
Pretended Networking ID Filter 1 Register (CAN_FLT_ID1)	S/U	Yes	Yes
Pretended Networking DLC Filter register (CAN_FLT_DLC)	S/U	Yes	Yes
Pretended Networking Payload Low Filter 1 register (CAN_PL1_LO)	S/U	Yes	Yes
Pretended Networking Payload High Filter 1 register (CAN_PL1_HI)	S/U	Yes	Yes
Pretended Networking ID Filter 2 Register / ID Mask register (CAN_FLT_ID2_IDMASK)	S/U	Yes	Yes
Pretended Networking Payload Low Filter 2 Register / Payload Low Mask Register (CAN_PL2_PLMASK_LO)	S/U	Yes	Yes
Pretended Networking Payload High Filter 2 Register / Payload High Mask Register (CAN_PL2_PLMASK_HI)	S/U	Yes	Yes
Pretended Networking Wake Up Message Buffer 0 register (CAN_WMB0)	S/U	Yes	No
Pretended Networking Wake Up Message Buffer 1 register (CAN_WMB1)	S/U	Yes	No
Pretended Networking Wake Up Message Buffer 2 register (CAN_WMB2)	S/U	Yes	No

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-7. Register access and reset information (continued)

Register	Access type	Affected by hard reset	Affected by soft reset
Pretended Networking Wake Up Message Buffer 3 register (CAN_WMB3)	S/U	Yes	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.

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).

CAN memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Module Configuration Register (CAN_MCR)	32	R/W	See section	46.4.2/1190
4	Control 1 register (CAN_CTRL1)	32	R/W	0000_0000h	46.4.3/1195
8	Free Running Timer (CAN_TIMER)	32	R/W	0000_0000h	46.4.4/1199
10	Rx Mailboxes Global Mask Register (CAN_RXMGMASK)	32	R/W	Undefined	46.4.5/1200
14	Rx 14 Mask register (CAN_RX14MASK)	32	R/W	Undefined	46.4.6/1201
18	Rx 15 Mask register (CAN_RX15MASK)	32	R/W	Undefined	46.4.7/1202
1C	Error Counter (CAN_ECR)	32	R/W	0000_0000h	46.4.8/1203
20	Error and Status 1 register (CAN_ESR1)	32	R/W	See section	46.4.9/1205
28	Interrupt Masks 1 register (CAN_IMASK1)	32	R/W	0000_0000h	46.4.10/ 1212
30	Interrupt Flags 1 register (CAN_IFLAG1)	32	R/W	0000_0000h	46.4.11/ 1212
34	Control 2 register (CAN_CTRL2)	32	R/W	See section	46.4.12/ 1215
38	Error and Status 2 register (CAN_ESR2)	32	R/W	0000_0000h	46.4.13/ 1219
44	CRC Register (CAN_CRCR)	32	R	0000_0000h	46.4.14/ 1220
48	Rx FIFO Global Mask register (CAN_RXFGMASK)	32	R/W	Undefined	46.4.15/ 1221
4C	Rx FIFO Information Register (CAN_RXFIR)	32	R	Undefined	46.4.16/ 1222
50	CAN Bit Timing Register (CAN_CBT)	32	R/W	See section	46.4.17/ 1223

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
880	Rx Individual Mask Registers (CAN_RXIMR0)	32	R/W	Undefined	46.4.18/ 1224
884	Rx Individual Mask Registers (CAN_RXIMR1)	32	R/W	Undefined	46.4.18/ 1224
888	Rx Individual Mask Registers (CAN_RXIMR2)	32	R/W	Undefined	46.4.18/ 1224
88C	Rx Individual Mask Registers (CAN_RXIMR3)	32	R/W	Undefined	46.4.18/ 1224
890	Rx Individual Mask Registers (CAN_RXIMR4)	32	R/W	Undefined	46.4.18/ 1224
894	Rx Individual Mask Registers (CAN_RXIMR5)	32	R/W	Undefined	46.4.18/ 1224
898	Rx Individual Mask Registers (CAN_RXIMR6)	32	R/W	Undefined	46.4.18/ 1224
89C	Rx Individual Mask Registers (CAN_RXIMR7)	32	R/W	Undefined	46.4.18/ 1224
8A0	Rx Individual Mask Registers (CAN_RXIMR8)	32	R/W	Undefined	46.4.18/ 1224
8A4	Rx Individual Mask Registers (CAN_RXIMR9)	32	R/W	Undefined	46.4.18/ 1224
8A8	Rx Individual Mask Registers (CAN_RXIMR10)	32	R/W	Undefined	46.4.18/ 1224
8AC	Rx Individual Mask Registers (CAN_RXIMR11)	32	R/W	Undefined	46.4.18/ 1224
8B0	Rx Individual Mask Registers (CAN_RXIMR12)	32	R/W	Undefined	46.4.18/ 1224
8B4	Rx Individual Mask Registers (CAN_RXIMR13)	32	R/W	Undefined	46.4.18/ 1224
8B8	Rx Individual Mask Registers (CAN_RXIMR14)	32	R/W	Undefined	46.4.18/ 1224
8BC	Rx Individual Mask Registers (CAN_RXIMR15)	32	R/W	Undefined	46.4.18/ 1224
B00	Pretended Networking Control 1 Register (CAN_CTRL1_PN)	32	R/W	See section	46.4.19/ 1226
B04	Pretended Networking Control 2 Register (CAN_CTRL2_PN)	32	R/W	0000_0000h	46.4.20/ 1227
B08	Pretended Networking Wake Up Match Register (CAN_WU_MTC)	32	R/W	See section	46.4.21/ 1228
B0C	Pretended Networking ID Filter 1 Register (CAN_FLT_ID1)	32	R/W	See section	46.4.22/ 1230
B10	Pretended Networking DLC Filter Register (CAN_FLT_DLC)	32	R/W	0000_0008h	46.4.23/ 1231
B14	Pretended Networking Payload Low Filter 1 Register (CAN_PL1_LO)	32	R/W	0000_0000h	46.4.24/ 1232

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
B18	Pretended Networking Payload High Filter 1 Register (CAN_PL1_HI)	32	R/W	0000_0000h	46.4.25/ 1232
B1C	Pretended Networking ID Filter 2 Register / ID Mask Register (CAN_FLT_ID2_IDMASK)	32	R/W	See section	46.4.26/ 1233
B20	Pretended Networking Payload Low Filter 2 Register / Payload Low Mask Register (CAN_PL2_PLMASK_LO)	32	R/W	0000_0000h	46.4.27/ 1234
B24	Pretended Networking Payload High Filter 2 low order bits / Payload High Mask Register (CAN_PL2_PLMASK_HI)	32	R/W	0000_0000h	46.4.28/ 1235
B40	Wake Up Message Buffer Register for C/S (CAN_WMB0_CS)	32	R	See section	46.4.29/ 1235
B44	Wake Up Message Buffer Register for ID (CAN_WMB0_ID)	32	R	0000_0000h	46.4.30/ 1236
B48	Wake Up Message Buffer Register for Data 0-3 (CAN_WMB0_D03)	32	R	0000_0000h	46.4.31/ 1237
B4C	Wake Up Message Buffer Register Data 4-7 (CAN_WMB0_D47)	32	R	0000_0000h	46.4.32/ 1238
B50	Wake Up Message Buffer Register for C/S (CAN_WMB1_CS)	32	R	See section	46.4.29/ 1235
B54	Wake Up Message Buffer Register for ID (CAN_WMB1_ID)	32	R	0000_0000h	46.4.30/ 1236
B58	Wake Up Message Buffer Register for Data 0-3 (CAN_WMB1_D03)	32	R	0000_0000h	46.4.31/ 1237
B5C	Wake Up Message Buffer Register Data 4-7 (CAN_WMB1_D47)	32	R	0000_0000h	46.4.32/ 1238
B60	Wake Up Message Buffer Register for C/S (CAN_WMB2_CS)	32	R	See section	46.4.29/ 1235
B64	Wake Up Message Buffer Register for ID (CAN_WMB2_ID)	32	R	0000_0000h	46.4.30/ 1236
B68	Wake Up Message Buffer Register for Data 0-3 (CAN_WMB2_D03)	32	R	0000_0000h	46.4.31/ 1237
B6C	Wake Up Message Buffer Register Data 4-7 (CAN_WMB2_D47)	32	R	0000_0000h	46.4.32/ 1238
B70	Wake Up Message Buffer Register for C/S (CAN_WMB3_CS)	32	R	See section	46.4.29/ 1235
B74	Wake Up Message Buffer Register for ID (CAN_WMB3_ID)	32	R	0000_0000h	46.4.30/ 1236
B78	Wake Up Message Buffer Register for Data 0-3 (CAN_WMB3_D03)	32	R	0000_0000h	46.4.31/ 1237
B7C	Wake Up Message Buffer Register Data 4-7 (CAN_WMB3_D47)	32	R	0000_0000h	46.4.32/ 1238
C00	CAN FD Control Register (CAN_FDCTRL)	32	R/W	See section	46.4.33/ 1238
C04	CAN FD Bit Timing Register (CAN_FDCBT)	32	R/W	See section	46.4.34/ 1242

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

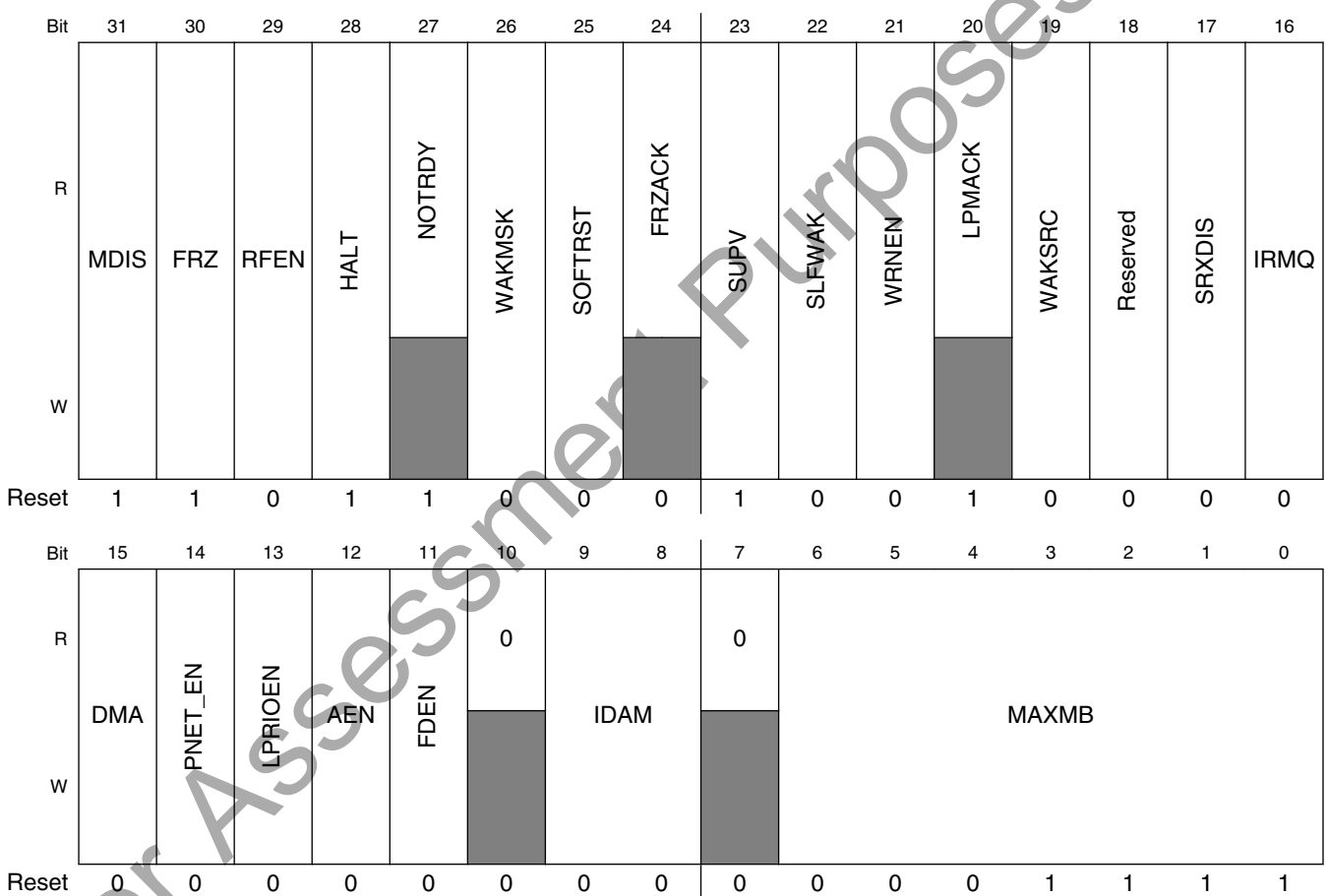
CAN memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
C08	CAN FD CRC Register (CAN_FDCRC)	32	R	0000_0000h	46.4.35/ 1244

46.4.2 Module Configuration Register (CAN_MCR)

This register defines global system configurations, such as the module operation modes and the maximum message buffer configuration.

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



CAN_MCR field descriptions

Field	Description
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.

Table continues on the next page...

For Assessment Purposes Only

CAN_MCR field descriptions (continued)

Field	Description
	0 Enable the FlexCAN module. 1 Disable the FlexCAN module.
30 FRZ	Freeze Enable The FRZ bit specifies the FlexCAN behavior when the HALT bit in the CAN_MCR Register 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. 0 Not enabled to enter Freeze mode. 1 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). 0 Rx FIFO not enabled. 1 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 CAN_CTRL1 and CAN_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. 0 No Freeze mode request. 1 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, Stop mode or Freeze mode. It is negated once FlexCAN has exited these modes. This bit is not affected by soft reset. 0 FlexCAN module is either in Normal mode, Listen-Only mode or Loop-Back mode. 1 FlexCAN module is either in Disable 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. 0 Wake Up Interrupt is disabled. 1 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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_MCR field descriptions (continued)

Field	Description
	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. 0 No reset request. 1 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 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 Section "Protocol Timing"). 0 FlexCAN not in Freeze mode, prescaler running. 1 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. 0 FlexCAN is in User mode. Affected registers allow both Supervisor and Unrestricted accesses. 1 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 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. When Pretended Networking mode is set, this feature must be disabled. 0 FlexCAN Self Wake Up feature is disabled. 1 FlexCAN Self Wake Up feature is enabled.
21 WRNEN	Warning Interrupt Enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_MCR field descriptions (continued)

Field	Description
	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. 0 TWRNINT and RWRNINT bits are zero, independent of the values in the error counters. 1 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, 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 Section "Protocol Timing"). When FlexCAN is in Pretended Networking mode LPMACK will be negated within 180 CAN bits after the low-power mode request removal. 0 FlexCAN is not in a low-power mode. 1 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. 0 FlexCAN uses the unfiltered Rx input to detect recessive to dominant edges on the CAN bus. 1 FlexCAN uses the filtered Rx input to detect recessive to dominant edges on the CAN bus.
18 Reserved	This field is reserved. When writing to this field, always write the reset value.
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. 0 Self reception enabled. 1 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 CAN_RXMGMASK, CAN_RX14MASK, CAN_RX15MASK and CAN_RXFGMASK. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0 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. 1 Individual Rx masking and queue feature are enabled.
15 DMA	DMA Enable

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_MCR field descriptions (continued)

Field	Description
	The DMA Enable bit controls whether the DMA feature is enabled or not. The DMA feature can only be used in Rx FIFO, consequently the bit CAN_MCR[RFEN] must be asserted. When DMA and RFEN are set, the 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. 0 DMA feature for RX FIFO disabled. 1 DMA feature for RX FIFO enabled.
14 PNET_EN	Pretended Networking Enable This bit enables the Pretended Networking functionality. Once in Doze mode or Stop mode, CAN_PE sub-block is kept operational, able to process Rx message filtering as defined by the Pretended Networking configuration registers. See Receive Process under Pretended Networking Mode. This bit can be written in Freeze mode only. 0 Pretended Networking mode is disabled. 1 Pretended Networking mode is enabled.
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. 0 Local Priority disabled. 1 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 the CAN_MCR[AEN] is asserted. 0 Abort disabled. 1 Abort enabled.
11 FDEN	CAN FD operation enable This bit enables the CAN with Flexible Data rate (CAN FD) operation. This bit can be written in Freeze mode only. **NOTE:** The Rx FIFO Enable (RFEN) bit cannot be set if FDEN is asserted. 1 CAN FD is enabled. FlexCAN is able to receive and transmit messages in both CAN FD and CAN 2.0 formats. 0 CAN FD is disabled. FlexCAN is able to receive and transmit messages in CAN 2.0 format.
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 IDAM	ID Acceptance Mode

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_MCR field descriptions (continued)

Field	Description
	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. 00 Format A: One full ID (standard and extended) per ID Filter Table element. 01 Format B: Two full standard IDs or two partial 14-bit (standard and extended) IDs per ID Filter Table element. 10 Format C: Four partial 8-bit Standard IDs per ID Filter Table element. 11 Format D: All frames rejected.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 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 RFFN bit in CAN_CTRL2 register. 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).

For Assessment

46.4.3 Control 1 register (CAN_CTRL1)

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 CAN_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 CAN_CTRL1 register for CAN bit timing. Instead use the the CAN_CBT register's EPRESDIV, ERJW, EPSEG1, EPSEG2, and EPROPSEG fields.

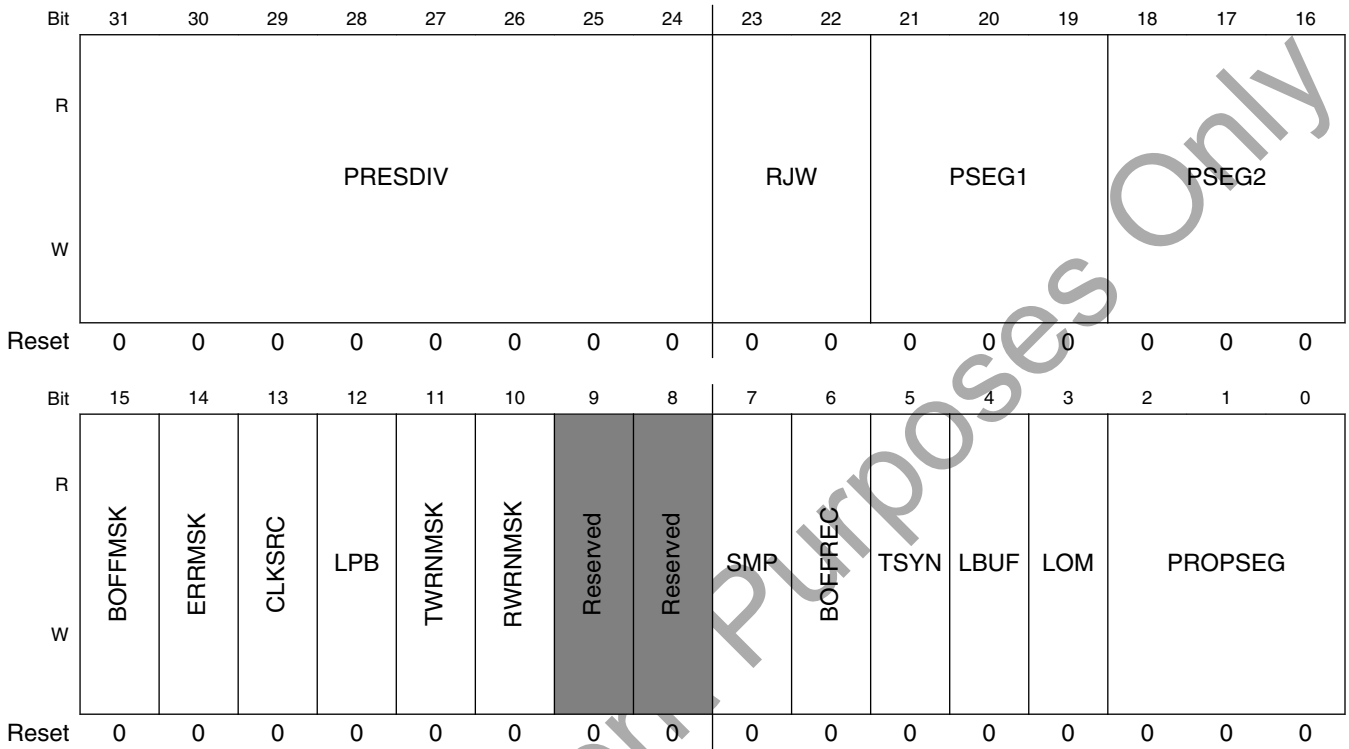
The contents of this register are not affected by soft reset.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

NOTE

The CAN bit variables in CAN_CTRL1 and in CAN_CBT are stored in the same register.

Address: 0h base + 4h offset = 4h



Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_CTRL1 field descriptions

Field	Description
31–24 PRESDIV	Prescaler Division Factor 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 Section "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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_CTRL1 field descriptions (continued)

Field	Description
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 BOFFINT in CAN_ESR1 register. 0 Bus Off interrupt disabled. 1 Bus Off interrupt enabled.
14 ERRMSK	Error Interrupt Mask This bit provides a mask for the Error Interrupt ERRINT in the CAN_ESR1 register. 0 Error interrupt disabled. 1 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". 0 The CAN engine clock source is the oscillator clock. Under this condition, the oscillator clock frequency must be lower than the bus clock. 1 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. **NOTE:** In this mode, the CAN_MCR[SRXDIS] cannot be asserted because this will impede the self reception of a transmitted message. **NOTE:** The TDCEN bit in CAN_FDCTRL register must be disabled when LPB is asserted. 0 Loop Back disabled. 1 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] bit is negated. This bit can be written only if CAN_MCR[WRNEN] bit is asserted.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_CTRL1 field descriptions (continued)

Field	Description
	0 Tx Warning Interrupt disabled. 1 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 CAN_MCR[WRNEN] bit is negated. This bit can be written only if CAN_MCR[WRNEN] bit is asserted. 0 Rx Warning Interrupt disabled. 1 Rx Warning Interrupt enabled.
9 Reserved	This field is reserved.
8 Reserved	This field is 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). 0 Just one sample is used to determine the bit value. 1 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. 0 Automatic recovering from Bus Off state enabled. 1 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 the RFEN bit in CAN_MCR 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. 0 Timer Sync feature disabled 1 Timer Sync feature enabled
4 LBUF	Lowest Buffer Transmitted First

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_CTRL1 field descriptions (continued)

Field	Description
	This bit defines the ordering mechanism for Message Buffer transmission. When asserted, the CAN_MCR[LPRIOEN] bit does not affect the priority arbitration. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0 Buffer with highest priority is transmitted first. 1 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 CAN_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 (RXERRCNT) in CAN_ECR register, 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. 0 Listen-Only mode is deactivated. 1 FlexCAN module operates in Listen-Only mode.
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.

For Assessment

46.4.4 Free Running Timer (CAN_TIMER)

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 the TIMER_SRC in CAN_CTRL2 register 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 the TIMER_SRC bit in CAN_CTRL2 register 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, Stop, Pretended Networking and Freeze modes.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

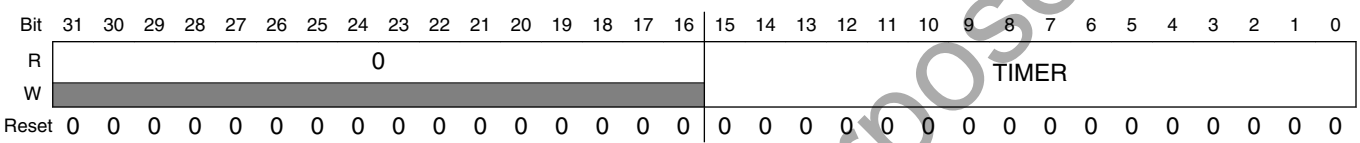
The timer value is captured when the second bit of the identifier field of any frame is on the CAN bus. This captured value is written into the Time Stamp entry in a message buffer after a successful reception or transmission of a message.

If bit CAN_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".

Address: 0h base + 8h offset = 8h



CAN_TIMER field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TIMER	Timer Value Contains the free-running counter value.

46.4.5 Rx Mailboxes Global Mask Register (CAN_RXMGMASK)

This register is located in RAM.

RXMGMASK is provided for legacy application support.

- When the CAN_MCR[IRMQ] bit is negated, RXMGMASK is always in effect (the bits in the MG field will mask the Mailbox filter bits).
- When the CAN_MCR[IRMQ] bit 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.

For Assessment Purposes Only

Address: 0h base + 10h offset = 10h

Bit	31	30	29	28	27	26	25	24	23	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	MG																															
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.

CAN_RXMGMASK field descriptions

Field	Description
MG	Rx Mailboxes Global Mask Bits 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.

SMB[RTR] [1]	CAN_CTRL2[RRS]	CAN_CTRL2[EACEN]	Mailbox filter fields			
			MB[RTR]	MB[IDE]	MB[ID]	Reserved
0	-	0	note [2]	note [3]	MG[28:0]	MG[31:29]
0	-	1	MG[31]	MG[30]	MG[28:0]	MG[29]
1	0	-	-	-	-	MG[31:0]
1	1	0	-	-	MG[28:0]	MG[31:29]
1	1	1	MG[31]	MG[30]	MG[28:0]	MG[29]

1. RTR bit of the Incoming Frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).
2. If the CTRL2[EACEN] bit is negated, the RTR bit of Mailbox is never compared with the RTR bit of the incoming frame.
3. If the CAN_CTRL2[EACEN] bit is negated, the IDE bit of Mailbox is always compared with the IDE bit of the incoming frame.

0 The corresponding bit in the filter is "don't care."
1 The corresponding bit in the filter is checked.

1. RTR bit of the Incoming Frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).
2. If the CTRL2[EACEN] bit is negated, the RTR bit of Mailbox is never compared with the RTR bit of the incoming frame.
3. If the CAN_CTRL2[EACEN] bit is negated, the IDE bit of Mailbox is always compared with the IDE bit of the incoming frame.

46.4.6 Rx 14 Mask register (CAN_RX14MASK)

This register is located in RAM.

RX14MASK is provided for legacy application support. When the CAN_MCR[IRMQ] bit is asserted, RX14MASK has no effect.

RX14MASK is used to mask the filter fields of Message Buffer 14.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

This register can only be programmed while the module is in Freeze mode as it is blocked by hardware in other modes.

Address: 0h base + 14h offset = 14h

Bit	31	30	29	28	27	26	25	24	23	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	RX14M																															
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.

CAN_RX14MASK field descriptions

Field	Description
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 CAN_RXMGMASK register. 0 The corresponding bit in the filter is "don't care." 1 The corresponding bit in the filter is checked.

46.4.7 Rx 15 Mask register (CAN_RX15MASK)

This register is located in RAM.

RX15MASK is provided for legacy application support. When the CAN_MCR[IRMQ] bit 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.

Address: 0h base + 18h offset = 18h

Bit	31	30	29	28	27	26	25	24	23	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	RX15M																															
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.

CAN_RX15MASK field descriptions

Field	Description
RX15M	Rx Buffer 15 Mask Bits

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_RX15MASK field descriptions (continued)

Field	Description
	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. 0 The corresponding bit in the filter is "don't care." 1 The corresponding bit in the filter is checked.

46.4.8 Error Counter (CAN_ECR)

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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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.
- When FlexCAN is in Pretended Networking mode, the RXERRCNT and RXERRCNT_FAST keep counting errors and error flags are stored. The TXERRCNT and TXERRCNT_FAST preserve their values and do not change, since no transmission occurs under Pretended Networking mode. Error counters and error flags that changed values while in Pretended Networking mode are updated in CAN_ECR and CAN_ESR1 when FlexCAN resumes back to Normal mode. The FAST error flags in CAN_ESR1 register will not be set if FlexCAN is in Pretended Networking mode.

Address: 0h base + 1Ch offset = 1Ch

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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_ECR field descriptions

Field	Description
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.
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.
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.

46.4.9 Error and Status 1 register (CAN_ESR1)

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 cummulative 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.

For Assessment Purp

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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

Address: 0h base + 20h offset = 20h

Bit	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

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BIT1ERR	BIT0ERR	ACKERR	CRCERR	FRMERR	STFERR	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

CAN_ESR1 field descriptions

Field	Description
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. 0 No such occurrence. 1 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. 0 No such occurrence. 1 At least one bit sent as dominant is received as recessive.
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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. 0 No such occurrence. 1 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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_ESR1 field descriptions (continued)

Field	Description
	0 No such occurrence. 1 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. 0 No such occurrence. 1 A Stuffing Error occurred since last read of this register.
25–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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. 0 Overrun has not occurred. 1 Overrun has occured.
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. 0 No such occurrence. 1 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. 0 No such occurrence. 1 FlexCAN module has completed Bus Off process.
18 SYNCH	CAN Synchronization Status 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. 0 FlexCAN is not synchronized to the CAN bus. 1 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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_ESR1 field descriptions (continued)

Field	Description
	When FlexCAN returns to Normal mode from Pretended Network mode (see Receive Process under Pretended Networking Mode), this bit is not updated. 0 No such occurrence. 1 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. When FlexCAN returns to Normal mode from Pretended Network mode (see Receive Process under Pretended Networking Mode), this bit is updated to reflect the Rx error counter state. 0 No such occurrence. 1 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. This bit is updated when FlexCAN returns to Normal mode from Pretended Network mode. **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. 0 No such occurrence. 1 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. This bit is updated when FlexCAN returns to Normal mode from Pretended Network mode. 0 No such occurrence. 1 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. This bit is updated when FlexCAN returns to Normal mode from Pretended Network mode. 0 No such occurrence. 1 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. This bit is updated when FlexCAN returns to Normal mode from Pretended Network mode.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_ESR1 field descriptions (continued)

Field	Description
	0 No such occurrence. 1 A CRC error occurred since last read of this register.
11 FRMERR	Form Error 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. This bit is updated when FlexCAN returns to Normal mode from Pretended Network mode. 0 No such occurrence. 1 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. This bit is updated when FlexCAN returns to Normal mode from Pretended Network mode. 0 No such occurrence. 1 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. 0 No such occurrence. 1 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. Additionally, it is updated when FlexCAN returns to Normal mode from Pretended Networking mode. 0 No such occurrence. 1 RXERRCNT is greater than or equal to 96.
7 IDLE	This bit indicates when CAN bus is in IDLE state. See the table in the overall CAN_ESR1 register description. 0 No such occurrence. 1 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. 0 FlexCAN is not transmitting a message. 1 FlexCAN is transmitting a message.
5–4 FLTCONF	Fault Confinement State This 2-bit field indicates the Confinement State of the FlexCAN module.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_ESR1 field descriptions (continued)

Field	Description
	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". 00 Error Active 01 Error Passive 1x 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. 0 FlexCAN is not receiving a message. 1 FlexCAN is receiving a message.
2 BOFFINT	Bus Off Interrupt 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. 0 No such occurrence. 1 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. 0 No such occurrence. 1 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: • 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. 0 No such occurrence. 1 Indicates a recessive to dominant transition was received on the CAN bus.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

46.4.10 Interrupt Masks 1 register (CAN_IMASK1)

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.

Address: 0h base + 28h offset = 28h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF31TO0M																															
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

CAN_IMASK1 field descriptions

Field	Description
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. 0 The corresponding buffer Interrupt is disabled. 1 The corresponding buffer Interrupt is enabled.

46.4.11 Interrupt Flags 1 register (CAN_IFLAG1)

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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

Address: 0h base + 30h offset = 30h

Bit	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

Bit	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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_IFLAG1 field descriptions

Field	Description
31–8 BUF31TO8I	Buffer MB_i Interrupt Each bit flags the corresponding FlexCAN Message Buffer interrupt for MB31 to MB8. 0 The corresponding buffer has no occurrence of successfully completed transmission or reception. 1 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. 0 No occurrence of MB7 completing transmission/reception when MCR[RFEN]=0, or of Rx FIFO overflow when MCR[RFEN]=1 1 MB7 completed transmission/reception when MCR[RFEN]=0, or Rx FIFO overflow when 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. 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. 0 No occurrence of MB6 completing transmission/reception when MCR[RFEN]=0, or of Rx FIFO almost full when MCR[RFEN]=1 1 MB6 completed transmission/reception when MCR[RFEN]=0, or Rx FIFO almost full when 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 MCR[RFEN] is changed by CPU writes. When 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 MCR[DMA] bit is enabled, this flag generates a DMA request and the CPU must not clear this bit by writing 1 in BUF5I. 0 No occurrence of MB5 completing transmission/reception when MCR[RFEN]=0, or of frame(s) available in the FIFO, when MCR[RFEN]=1 1 MB5 completed transmission/reception when MCR[RFEN]=0, or frame(s) available in the Rx FIFO when MCR[RFEN]=1. It generates a DMA request in case of MCR[RFEN] and MCR[DMA] are enabled.
4–1 BUF4TO1I	Buffer MB_i Interrupt Or "reserved"

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_IFLAG1 field descriptions (continued)

Field	Description
	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. 0 The corresponding buffer has no occurrence of successfully completed transmission or reception when MCR[RFEN]=0. 1 The corresponding buffer has successfully completed transmission or reception when 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 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. 0 The corresponding buffer has no occurrence of successfully completed transmission or reception when MCR[RFEN]=0. 1 The corresponding buffer has successfully completed transmission or reception when MCR[RFEN]=0.

46.4.12 Control 2 register (CAN_CTRL2)

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.

Address: 0h base + 34h offset = 34h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERRMSK_FAST	BOFFDONEMSK	0	0	RFFN				TASD					MRP	RRS	EACEN
W																
Reset	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0

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

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_CTRL2 field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>31
ERRMSK_FAST</td><td>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.

0 ERRINT_FAST Error interrupt disabled.
1 ERRINT_FAST Error interrupt enabled.</td></tr>
<tr><td>30
BOFFDONEMSK</td><td>Bus Off Done Interrupt Mask

This bit provides a mask for the Bus Off Done Interrupt in CAN_ESR1 register.

0 Bus Off Done interrupt disabled.
1 Bus Off Done interrupt enabled.</td></tr>
<tr><td>29
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>28
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>27–24
RFFN</td><td>Number Of Rx FIFO Filters

This 4-bit field defines the number of Rx FIFO filters, as shown in the following table. 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

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.

<table>
<tr><th>RFFN[3:0]</th><th>Number of Rx FIFO filter elements</th><th>Message Buffers occupied by Rx FIFO and ID Filter Table</th><th>Remaining Available Mailboxes</th><th>Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks</th><th>Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask</th></tr>
<tr><td>0x0</td><td>8</td><td>MB 0-7</td><td>MB 8-63</td><td>Elements 0-7</td><td>none</td></tr>
<tr><td>0x1</td><td>16</td><td>MB 0-9</td><td>MB 10-63</td><td>Elements 0-9</td><td>Elements 10-15</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_CTRL2 field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>RFFN[3:0]</th><th>Number of Rx FIFO filter elements</th><th>Message Buffers occupied by Rx FIFO and ID Filter Table</th><th>Remaining Available Mailboxes</th><th>Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks</th><th>Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask</th></tr>
<tr><td>0x2</td><td>24</td><td>MB 0-11</td><td>MB 12-63</td><td>Elements 0-11</td><td>Elements 12-23</td></tr>
<tr><td>0x3</td><td>32</td><td>MB 0-13</td><td>MB 14-63</td><td>Elements 0-13</td><td>Elements 14-31</td></tr>
<tr><td>0x4</td><td>40</td><td>MB 0-15</td><td>MB 16-63</td><td>Elements 0-15</td><td>Elements 16-39</td></tr>
<tr><td>0x5</td><td>48</td><td>MB 0-17</td><td>MB 18-63</td><td>Elements 0-17</td><td>Elements 18-47</td></tr>
<tr><td>0x6</td><td>56</td><td>MB 0-19</td><td>MB 20-63</td><td>Elements 0-19</td><td>Elements 20-55</td></tr>
<tr><td>0x7</td><td>64</td><td>MB 0-21</td><td>MB 22-63</td><td>Elements 0-21</td><td>Elements 22-63</td></tr>
<tr><td>0x8</td><td>72</td><td>MB 0-23</td><td>MB 24-63</td><td>Elements 0-23</td><td>Elements 24-71</td></tr>
<tr><td>0x9</td><td>80</td><td>MB 0-25</td><td>MB 26-63</td><td>Elements 0-25</td><td>Elements 26-79</td></tr>
<tr><td>0xA</td><td>88</td><td>MB 0-27</td><td>MB 28-63</td><td>Elements 0-27</td><td>Elements 28-87</td></tr>
<tr><td>0xB</td><td>96</td><td>MB 0-29</td><td>MB 30-63</td><td>Elements 0-29</td><td>Elements 30-95</td></tr>
<tr><td>0xC</td><td>104</td><td>MB 0-31</td><td>MB 32-63</td><td>Elements 0-31</td><td>Elements 32-103</td></tr>
<tr><td>0xD</td><td>112</td><td>MB 0-33</td><td>MB 34-63</td><td>Elements 0-31</td><td>Elements 32-111</td></tr>
<tr><td>0xE</td><td>120</td><td>MB 0-35</td><td>MB 36-63</td><td>Elements 0-31</td><td>Elements 32-119</td></tr>
<tr><td>0xF</td><td>128</td><td>MB 0-37</td><td>MB 38-63</td><td>Elements 0-31</td><td>Elements 32-127</td></tr>
</table>
</td></tr>
<tr><td>23–19
TASD</td><td>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.</td></tr>
<tr><td>18
MRP</td><td>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.

0 Matching starts from Rx FIFO and continues on Mailboxes.
1 Matching starts from Mailboxes and continues on Rx FIFO.</td></tr>
<tr><td>17
RRS</td><td>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.

0 Remote Response Frame is generated.
1 Remote Request Frame is stored.</td></tr>
<tr><td>16
EACEN</td><td>Entire Frame Arbitration Field Comparison Enable For Rx Mailboxes</td></tr>
</table>

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_CTRL2 field descriptions (continued)

Field	Description
	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. 0 Rx Mailbox filter's IDE bit is always compared and RTR is never compared despite mask bits. 1 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. 0 The Free Running Timer is clocked by the CAN bit clock, which defines the baud rate on the CAN bus. 1 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. This field is writable only in Freeze mode. 0 Protocol Exception is disabled. 1 Protocol Exception is enabled.
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 STFCNTEN	Stuff Count Enable This bit enables the count of variable stuff bits inserted from the Start of Frame bit to the last bit of Data field. Also, the modulo 8 count of variable stuff bits plus the respective parity bit (even parity calculated over the 3-bit modulo 8 count) are combined as the 4-bit Stuff Count field and inserted before the CRC Sequence field. CRC calculation extends beyond the end of Data field and takes the Stuff Count field bits into account. This field is writable only in Freeze mode. 0 Stuff Count feature is disabled. Stuff Count bit field is not inserted before the CRC Sequence field. 1 Stuff Count feature is enabled. Stuff Count bit field is inserted before the CRC Sequence field.
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. 0 Edge Filter is enabled. 1 Edge Filter is disabled.
10–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	This field is reserved. When writing to this field, always write the reset value.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_CTRL2 field descriptions (continued)

Field	Description
0 Reserved	This field is reserved. When writing to this field, always write the reset value.

46.4.13 Error and Status 2 register (CAN_ESR2)

This register reports some general status information.

Address: 0h base + 38h offset = 38h

Bit	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

Bit	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

CAN_ESR2 field descriptions

Field	Description
31–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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	This field is reserved. This read-only field is reserved and always has the value 0.
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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Purposes Only

CAN_ESR2 field descriptions (continued)

Field	Description
	0 Contents of IMB and LPTM are invalid. 1 Contents of IMB and LPTM are valid.
13 IMB	Inactive Mailbox If 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. 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]. 0 If ESR2[VPS] is asserted, the ESR2[LPTM] is not an inactive Mailbox. 1 If ESR2[VPS] is asserted, there is at least one inactive Mailbox. LPTM content is the number of the first one.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

46.4.14 CRC Register (CAN_CRCR)

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.

Address: 0h base + 44h offset = 44h

Bit	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

Bit	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

For Assessment Purp

CAN_CRCR field descriptions

Field	Description
31–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–16 MBCRC	CRC Mailbox This field indicates the number of the Mailbox corresponding to the value in CAN_CRCR[TXCRC] field.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 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.

46.4.15 Rx FIFO Global Mask register (CAN_RXFGMASK)

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.

Address: 0h base + 48h offset = 48h

Bit	31	30	29	28	27	26	25	24	23	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	FGM																															
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.

CAN_RXFGMASK field descriptions

Field	Description
FGM	Rx FIFO Global Mask Bits These bits mask the ID Filter Table elements bits in a perfect alignment. The following table shows how the FGM bits correspond to each IDAF field.

Freescale Confidential Proprietary - Non-Disclosure Agreement required
Review Draft
Freescale Semiconductor, Inc.

CAN_RXFGMASK field descriptions (continued)

Field	Description

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.

0 The corresponding bit in the filter is "don't care."
1 The corresponding bit in the filter is checked.

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.

46.4.16 Rx FIFO Information Register (CAN_RXFIR)

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.

Address: 0h base + 4Ch offset = 4Ch

Bit	31	30	29	28	27	26	25	24	23	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																							IDHIT								
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.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_RXFIR field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 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.

46.4.17 CAN Bit Timing Register (CAN_CBT)

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].

Address: 0h base + 50h offset = 50h

Bit	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

Bit	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

CAN_CBT field descriptions

Field	Description
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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes O

CAN_CBT field descriptions (continued)

Field	Description
	0 Extended bit time definitions disabled. 1 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 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.
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.

46.4.18 Rx Individual Mask Registers (CAN_RXIMR*n*)

The RX Individual Mask Registers are used to store the acceptance masks for ID filtering in Rx MBs and the Rx FIFO.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

Address: 0h base + 880h offset + (4d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	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	MI																															
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.

CAN_RXIMR*n* field descriptions

Field	Description
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. 0 The corresponding bit in the filter is "don't care." 1 The corresponding bit in the filter is checked.

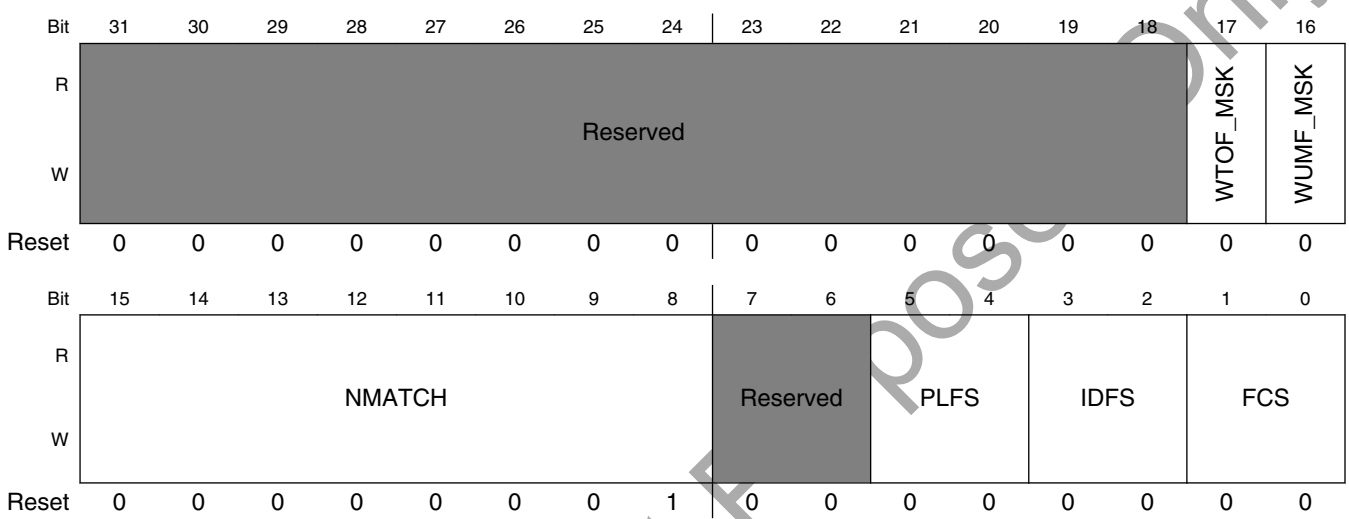
For Ass Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

46.4.19 Pretended Networking Control 1 Register (CAN_CTRL1_PN)

This register contains control bits for Pretended Networking mode filtering selection. Configure this register with the filter criteria to be used to receive wake up messages. It can be written in Freeze mode only, except WTOF_MSK and WUMF_MSK bits.

Address: 0h base + B00h offset = B00h



CAN_CTRL1_PN field descriptions

Field	Description
31–18 Reserved	This field is reserved.
17 WTOF_MSK	Wake Up by Timeout Flag Mask Bit This bit masks the generation of a wake up event originated by a timeout. 0 Timeout wake up event is disabled 1 Timeout wake up event is enabled
16 WUMF_MSK	Wake Up by Match Flag Mask Bit This bit masks the generation of a wake up event originated by a successful filtered Rx message. 0 Wake up match event is disabled 1 Wake up match event is enabled
15–8 NMATCH	Number of Messages Matching the Same Filtering Criteria This 8-bit field defines the number of times a given message must match the predefined filtering criteria for ID and/or PL before generating a wake up event. This quantity can be configured in the 1 to 255 range by using values from 0x01 to 0xFF, respectively. 00000001 Received message must match the predefined filtering criteria for ID and/or PL once before generating a wake up event. 00000010 Received message must match the predefined filtering criteria for ID and/or PL twice before generating a wake up event.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_CTRL1_PN field descriptions (continued)

Field	Description
	... ... 11111111 Received message must match the predefined filtering criteria for ID and/or PL 255 times before generating a wake up event.
7–6 Reserved	This field is reserved.
5–4 PLFS	Payload Filtering Selection This 2-bit field selects the level of payload filtering to be applied when FlexCAN is under Pretended Networking mode. Filtering does not accept remote messages (RTR=1) when payload filtering is active. 00 Match upon a payload contents against an exact target value 01 Match upon a payload value greater than or equal to a specified target value 10 Match upon a payload value smaller than or equal to a specified target value 11 Match upon a payload value inside a range, greater than or equal to a specified lower limit and smaller than or equal a specified upper limit
3–2 IDFS	ID Filtering Selection This 2-bit field selects the level of ID filtering to be applied when FlexCAN is under Pretended Networking mode. In ID filtering, the IDE and RTR bits are also considered as part of reception filter if IDE_MSK and RTR_MSK bits in the CAN_FLT_ID2_IDMASK register are set. 00 Match upon a ID contents against an exact target value 01 Match upon a ID value greater than or equal to a specified target value 10 Match upon a ID value smaller than or equal to a specified target value 11 Match upon a ID value inside a range, greater than or equal to a specified lower limit and smaller than or equal a specified upper limit
FCS	Filtering Combination Selection This 2-bit field selects the filtering criteria to be applied when FlexCAN is under Pretended Networking mode. See Receive Process under Pretended Networking Mode for more details. 00 Message ID filtering only 01 Message ID filtering and payload filtering 10 Message ID filtering occurring a specified number of times. 11 Message ID filtering and payload filtering a specified number of times

For Ass

46.4.20 Pretended Networking Control 2 Register (CAN_CTRL2_PN)

This register contains configuration bits for the timeout value under Pretended Networking mode. It can be written in Freeze mode only.

Address: 0h base + B04h offset = B04h

Bit	31	30	29	28	27	26	25	24	23	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																MATCHTO															
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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

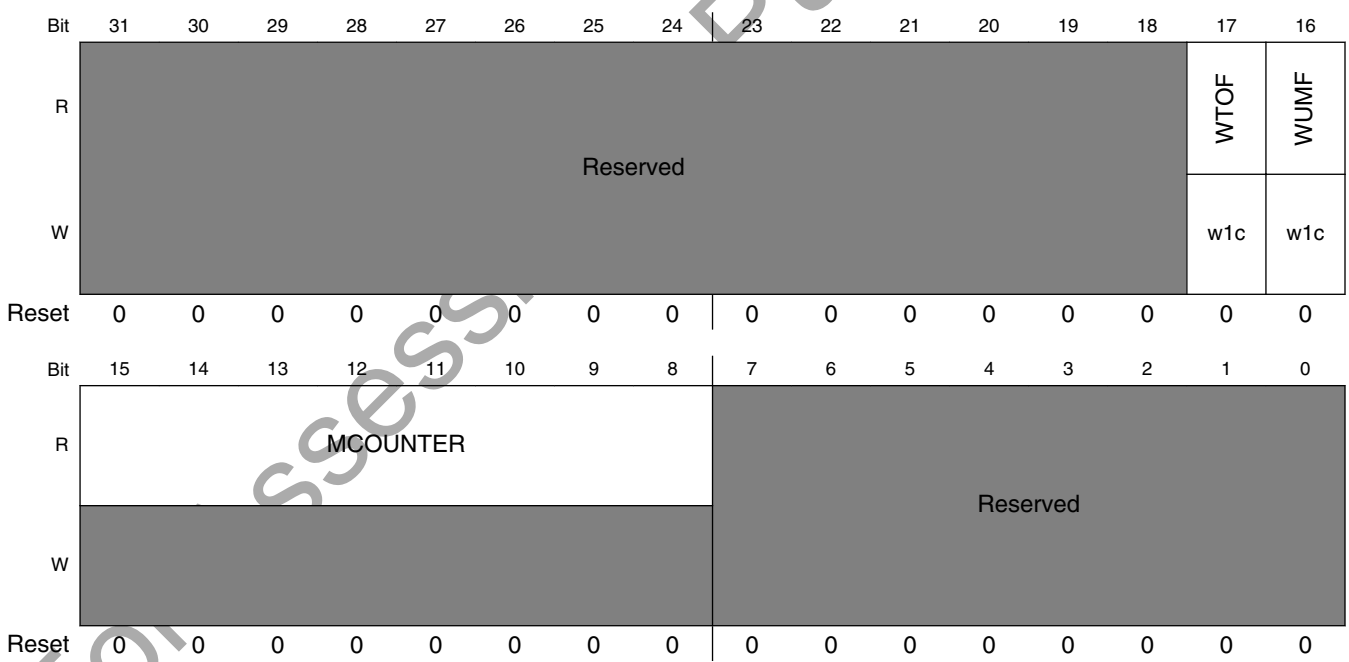
CAN_CTRL2_PN field descriptions

Field	Description
31–16 Reserved	This field is reserved.
MATCHTO	Timeout for No Message Matching the Filtering Criteria This 16-bit field defines a timeout value that generates a wake up event if CAN_MCR[PNET_EN] is asserted. If the timeout counter reaches the target value, when FlexCAN is under Pretended Networking mode, then a wake up event is generated. The timeout limit can be configured from 1 to 65535 to control an internal 16-bit up-count timer to produce a trigger upon reaching this configured value. The internal timer is incremented based on periodic time ticks, which period is 64 times the CAN Bit Time unit. When MATCHTO is configured with 0x0000 the timeout is disabled.

46.4.21 Pretended Networking Wake Up Match Register (CAN_WU_MTC)

This read-only register contains wake up information related to the matching processes performed while FlexCAN receives frames under Pretended Networking mode.

Address: 0h base + B08h offset = B08h



CAN_WU_MTC field descriptions

Field	Description
31–18 Reserved	This field is reserved.
17 WTOF	Wake Up by Timeout Flag Bit

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_WU_MTC field descriptions (continued)

Field	Description
	This bit identifies whether the FlexCAN has detected a timeout event during a time interval defined by CAN_CTRL2_PN[MATCHTO]. This flag generates a wake up event if CAN_CTRL1_PN[WTOF_MSK] is enabled. 0 No wake up by timeout event detected 1 Wake up by timeout event detected
16 WUMF	Wake Up by Match Flag Bit This bit identifies whether the FlexCAN has detected a matching Rx incoming message, which passed the filtering criteria specified in CAN_CTRL1_PN register. This flag generates a wake up event if CAN_CTRL1_PN[WUMF_MSK] is enabled. 0 No wake up by match event detected 1 Wake up by match event detected
15–8 MCOUNTER	Number of Matches while in Pretended Networking This 8-bit field reports the number of times a given message have been matched the predefined filtering criteria for ID and/or PL before a wake up event. This register is reset by FlexCAN when it enters in Pretended Networing mode, and is not affected by soft reset.
Reserved	This field is reserved.

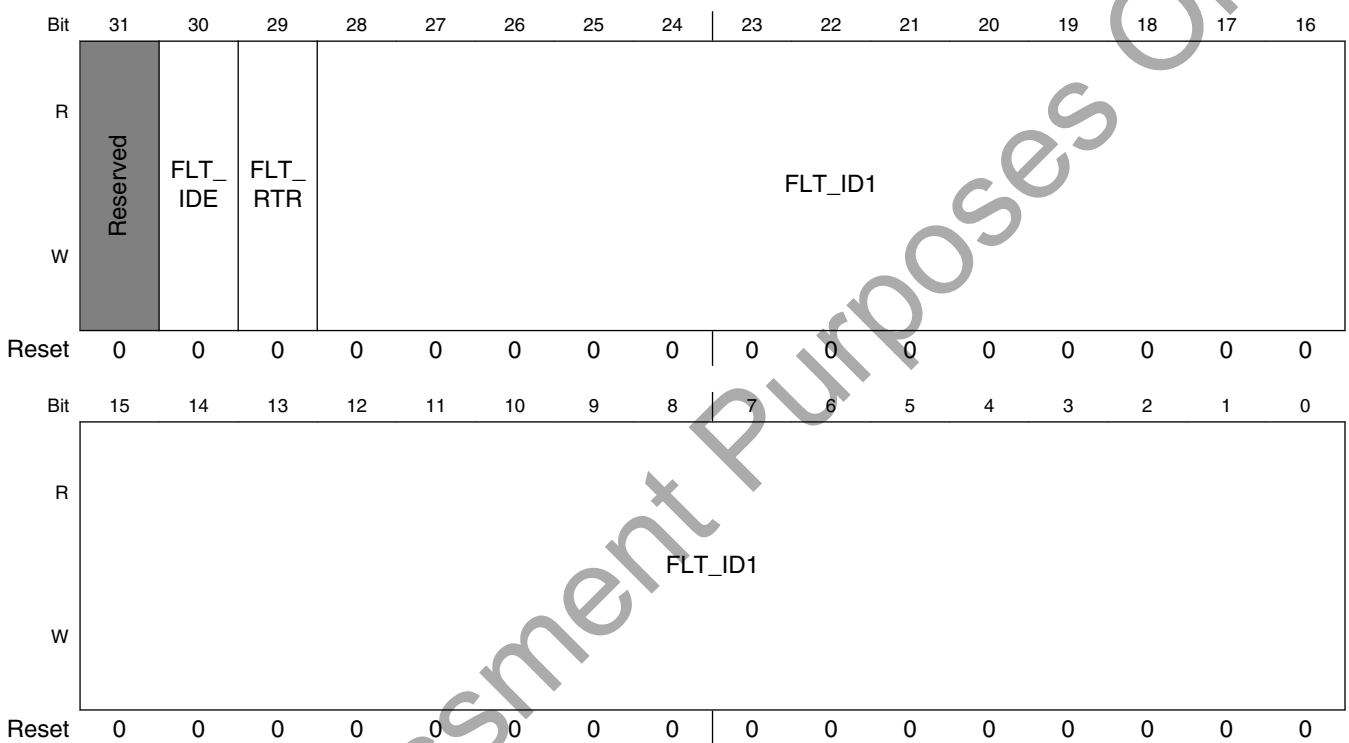
For Assessment Pur

Freescale Confidential Proprietary - Non-Disclosure Agreement required

46.4.22 Pretended Networking ID Filter 1 Register (CAN_FLT_ID1)

This register contains FLT_ID1 target value, as well as, IDE and RTR target values used to filter incoming message ID. The FLT_ID1 is used either for equal to, smaller than, greater than comparisons, or as the lower limit value in a ID range detection. It can be written in Freeze mode only.

Address: 0h base + B0Ch offset = B0Ch



ssment Purposes Only

CAN_FLT_ID1 field descriptions

Field	Description
31 Reserved	This field is reserved.
30 FLT_IDE	ID Extended Filter This bit identifies whether the frame format is standard or extended. It is used as part of the ID reception filter. 0 Accept standard frame format 1 Accept extended frame format
29 FLT_RTR	Remote Transmission Request Filter This bit identifies whether the frame is remote or not. It is used as part of the ID reception filter. 0 Reject remote frame (accept data frame) 1 Accept remote frame

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_FLT_ID1 field descriptions (continued)

Field	Description
FLT_ID1	ID Filter 1 for Pretended Networking filtering This 29-bit field defines either the 29 bits of a extended frame format, considering all bits, or the 11 bits of a standard frame format, considering just the 11 leftmost bits.

46.4.23 Pretended Networking DLC Filter Register (CAN_FLT_DLC)

This register contains the DLC inside range target values (FLT_DLC_LO and FLT_DLC_HI) used to filter incoming message. The DLC range is used only for payload filtering. It can be written in Freeze mode only.

NOTE

When a fixed quantity of data bytes is required, both FLT_DLC_LO and FLT_DLC_HI need to be configured with the same value, otherwise a range of DLC is considered for filtering (see Receive Process under Pretended Networking Mode).

Address: 0h base + B10h offset = B10h

Bit	31	30	29	28	27	26	25	24	23	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												FLT_DLC_LO				Reserved												FLT_DLC_HI			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

CAN_FLT_DLC field descriptions

Field	Description
31–20 Reserved	This field is reserved.
19–16 FLT_DLC_LO	Lower Limit for Length of Data Bytes Filter This field specifies the lower limit for the number of data bytes considered valid for payload comparison. It is used as part of payload reception filter.
15–4 Reserved	This field is reserved.
FLT_DLC_HI	Upper Limit for Length of Data Bytes Filter This field specifies the upper limit for the number of data bytes considered valid for payload comparison. It is used as part of payload reception filter.

ment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

46.4.24 Pretended Networking Payload Low Filter 1 Register (CAN_PL1_LO)

This register contains Payload Filter 1 low order bits of the target value used to filter incoming message payload. It is used either for “equal to”, “smaller than or equal”, “greater than or equal” comparisons, or as the lower limit value in a payload range detection. It can be written in Freeze mode only.

Address: 0h base + B14h offset = B14h

Bit	31	30	29	28	27	26	25	24	23	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	Data_byte_0								Data_byte_1								Data_byte_2								Data_byte_3							
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

CAN_PL1_LO field descriptions

Field	Description
31–24 Data_byte_0	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to the data byte 0.
23–16 Data_byte_1	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to the data byte 1.
15–8 Data_byte_2	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to the data byte 2.
Data_byte_3	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to the data byte 3.

46.4.25 Pretended Networking Payload High Filter 1 Register (CAN_PL1_HI)

This register contains Payload Filter 1 high order bits of the target value used to filter incoming message payload. It is used either for “equal to”, “smaller than or equal to”, “greater than or equal to” comparisons, or as the lower limit value in a payload range detection. It can be written in Freeze mode only.

Address: 0h base + B18h offset = B18h

Bit	31	30	29	28	27	26	25	24	23	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	Data_byte_4								Data_byte_5								Data_byte_6								Data_byte_7							
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

CAN_PL1_HI field descriptions

Field	Description
31–24 Data_byte_4	Payload Filter 1 high order bits for Pretended Networking payload filtering corresponding to the data byte 4.

Table continues on the next page...

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_PL1_HI field descriptions (continued)

Field	Description
23–16 Data_byte_5	Payload Filter 1 high order bits for Pretended Networking payload filtering corresponding to the data byte 5.
15–8 Data_byte_6	Payload Filter 1 high order bits for Pretended Networking payload filtering corresponding to the data byte 6.
Data_byte_7	Payload Filter 1 high order bits for Pretended Networking payload filtering corresponding to the data byte 7.

46.4.26 Pretended Networking ID Filter 2 Register / ID Mask Register (CAN_FLT_ID2_IDMASK)

This register contains FLT_ID2 target value used only as the upper limit value in ID range detection. Also, when exact ID filtering criteria is selected, this register is used to store the ID mask. IDE_MSK and RTR_MSK bits are used in both ID filtering (exact and range) to enable FLT_IDE and FLT_RTR respectively to be used as part of ID reception filter. This register can be written in Freeze mode only.

Address: 0h base + B1Ch offset = B1Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	IDE_MSK	RTR_MSK	FLT_ID2_IDMASK												
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	FLT_ID2_IDMASK															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

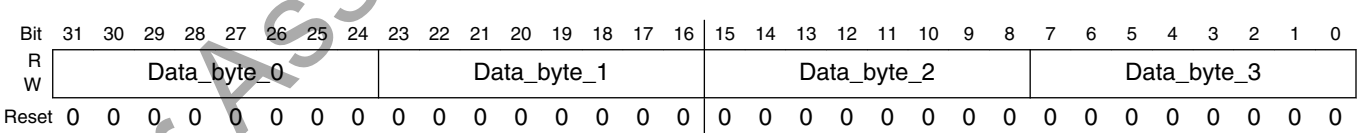
CAN_FLT_ID2_IDMASK field descriptions

Field	Description
31 Reserved	This field is reserved.
30 IDE_MSK	ID Extended Mask Bit This bit indicates whether the frame format (standard / extended) is used as part of the ID reception filter. 0 The corresponding bit in the filter is “don’t care” 1 The corresponding bit in the filter is checked
29 RTR_MSK	Remote Transmission Request Mask Bit This bit indicates whether the frame type (data / remote) is part of the ID reception filter. 0 The corresponding bit in the filter is “don’t care” 1 The corresponding bit in the filter is checked
FLT_ID2_ IDMASK	ID Filter 2 for Pretended Networking Filtering / ID Mask Bits for Pretended Networking ID Filtering This register is used to define either the ID filter value in extended frame format (29 bits), considering the FLT_ID2[28:0], or the ID filter value in standard frame format (11 bits), considering the FLT_ID2[28:18] (other bits in the [17:0] range have no meaning). Used only in range of ID filtering. It can also be used to define either the mask values for the extended frame format (29 bits), considering the IDMASK[28:0], or for the standard frame format (11 bits), considering the IDMASK[28:18] (other bits in the [17:0] range have no meaning). Used only in exact ID filtering.

46.4.27 Pretended Networking Payload Low Filter 2 Register / Payload Low Mask Register (CAN_PL2_PLMASK_LO)

This register has two functions. First, it contains the low order bits for the Payload Filter 2 used only as the upper limit value in a payload range detection. Second, when exact payload filtering criteria is selected, this register is used as payload mask. Otherwise, this register is unused. It can be written in Freeze mode only.

Address: 0h base + B20h offset = B20h



CAN_PL2_PLMASK_LO field descriptions

Field	Description
31–24 Data_byte_0	Payload Filter 2 low order bits / Payload Mask low order bits for Pretended Networking payload filtering corresponding to the data byte 0.
23–16 Data_byte_1	Payload Filter 2 low order bits / Payload Mask low order bits for Pretended Networking payload filtering corresponding to the data byte 1.
15–8 Data_byte_2	Payload Filter 2 low order bits / Payload Mask low order bits for Pretended Networking payload filtering corresponding to the data byte 2.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_PL2_PLMASK_LO field descriptions (continued)

Field	Description
Data_byte_3	Payload Filter 2 low order bits / Payload Mask low order bits for Pretended Networking payload filtering corresponding to the data byte 3.

46.4.28 Pretended Networking Payload High Filter 2 low order bits / Payload High Mask Register (CAN_PL2_PLMASK_HI)

This register has two functions. First, it contains the high order bits for the Payload Filter 2 used only as the upper limit value in a payload range detection. Second, when exact payload filtering criteria is selected, this register is used as payload mask. Otherwise, this register is unused. It can be written in Freeze mode only.

Address: 0h base + B24h offset = B24h

Bit	31	30	29	28	27	26	25	24	23	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	Data_byte_4								Data_byte_5								Data_byte_6								Data_byte_7							
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

CAN_PL2_PLMASK_HI field descriptions

Field	Description
31–24 Data_byte_4	Payload Filter 2 high order bits / Payload Mask high order bits for Pretended Networking payload filtering corresponding to the data byte 4.
23–16 Data_byte_5	Payload Filter 2 high order bits / Payload Mask high order bits for Pretended Networking payload filtering corresponding to the data byte 5.
15–8 Data_byte_6	Payload Filter 2 high order bits / Payload Mask high order bits for Pretended Networking payload filtering corresponding to the data byte 6.
Data_byte_7	Payload Filter 2 high order bits / Payload Mask high order bits for Pretended Networking payload filtering corresponding to the data byte 7.

46.4.29 Wake Up Message Buffer Register for C/S (CAN_WMB*n*_CS)

Each of the four WMBs contains a register to store the Control Status (C/S) information (IDE, RTR and DLC fields) of an incoming Rx message.

NOTE

The C/S registers are located at 0xB40 for WMB0, 0xB50 for WMB1, 0xB60 for WMB2, and 0xB70 for WMB3.

For Assessment Purposes Only

Address: 0h base + B40h offset + (16d × i), where i=0d to 3d

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

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

CAN_WMB*n*_CS field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22 SRR	Substitute Remote Request This bit can be received either recessive or dominant.
21 IDE	ID Extended Bit This bit identifies whether the frame format is standard or extended 0 Frame format is standard 1 Frame format is extended
20 RTR	Remote Transmission Request Bit This bit identifies whether the frame is remote or not. 0 Frame is data one (not remote) 1 Frame is a remote one
19–16 DLC	Length of Data in Bytes This 4-bit field is the length (in bytes) of the Rx data received when FlexCAN is in Pretended Networking mode. This 4-bit field is written by the FlexCAN module, copied from the DLC (Data Length Code) field of the received frame. The DLC field indicates which data bytes are valid.
Reserved	This field is reserved.

46.4.30 Wake Up Message Buffer Register for ID (CAN_WMB*n*_ID)

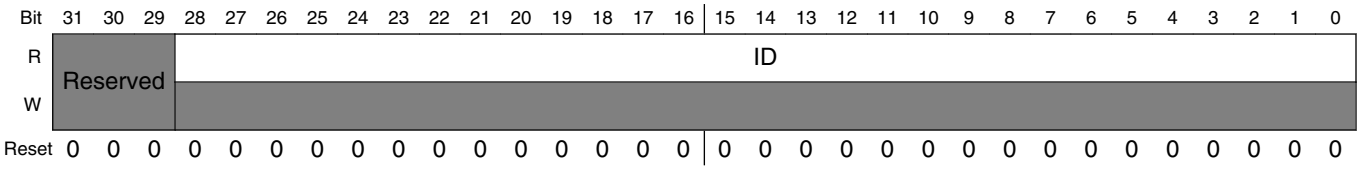
Each of the four WMBs contains a register to store the ID information of an incoming Rx message.

NOTE

The ID registers are located at 0xB44 for WMB0, 0xB54 for WMB1, 0xB64 for WMB2, and 0xB74 for WMB3.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Address: 0h base + B44h offset + (16d × i), where i=0d to 3d



CAN_WMB*n*_ID field descriptions

Field	Description
31–29 Reserved	This field is reserved.
ID	Received ID under Pretended Networking mode This register stores either the 29 bits of the extended frame format (considering the ID[28:0] field), or the 11 bits of the standard frame format (considering the ID[28:18] field only, the remaining bits in the ID[17:0] range have no meaning).

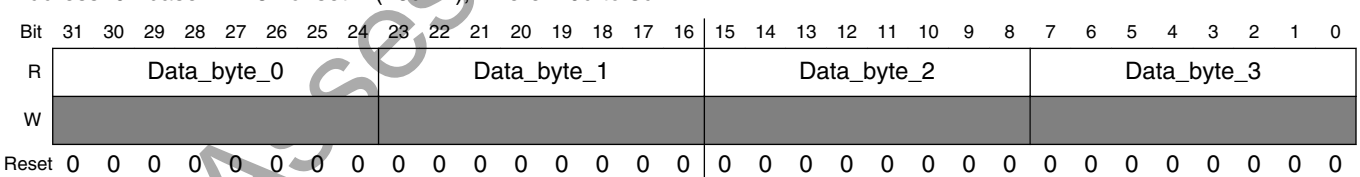
46.4.31 Wake Up Message Buffer Register for Data 0-3 (CAN_WMB*n*_D03)

Each of the four WMBs contains a register to store the data bytes 0 to 3 of the payload information of an incoming Rx message. This register's content is cleared when the incoming matched message is either a remote frame (RTR=1) or a data frame with DLC=0.

NOTE

The Data 0-3 registers are located at 0xB48 for WMB0, 0xB58 for WMB1, 0xB68 for WMB2, and 0xB78 for WMB3.

Address: 0h base + B48h offset + (16d × i), where i=0d to 3d



CAN_WMB*n*_D03 field descriptions

Field	Description
31–24 Data_byte_0	Received payload corresponding to the data byte 0 under Pretended Networking mode
23–16 Data_byte_1	Received payload corresponding to the data byte 1 under Pretended Networking mode
15–8 Data_byte_2	Received payload corresponding to the data byte 2 under Pretended Networking mode
Data_byte_3	Received payload corresponding to the data byte 3 under Pretended Networking mode

Assessment Purpos

Freescale Confidential Proprietary - Non-Disclosure Agreement required

46.4.32 Wake Up Message Buffer Register Data 4-7 (CAN_WMB*n*_D47)

Each of the four WMBs contains a register to store the data bytes 4 to 7 of the payload information of an incoming Rx message. This register's content is cleared when the incoming matched message is either a remote frame (RTR=1) or a data frame with DLC=0.

NOTE

The Data 4-7 registers are located at 0xB4C for WMB0, 0xB5C for WMB1, 0xB6C for WMB2, and 0xB7C for WMB3.

Address: 0h base + B4Ch offset + (16d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Data_byte_4								Data_byte_5								Data_byte_6								Data_byte_7							
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

CAN_WMB*n*_D47 field descriptions

Field	Description
31–24 Data_byte_4	Received payload corresponding to the data byte 4 under Pretended Networking mode
23–16 Data_byte_5	Received payload corresponding to the data byte 5 under Pretended Networking mode
15–8 Data_byte_6	Received payload corresponding to the data byte 6 under Pretended Networking mode
Data_byte_7	Received payload corresponding to the data byte 7 under Pretended Networking mode

46.4.33 CAN FD Control Register (CAN_FDCTRL)

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:

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-8. Number of Message Buffers

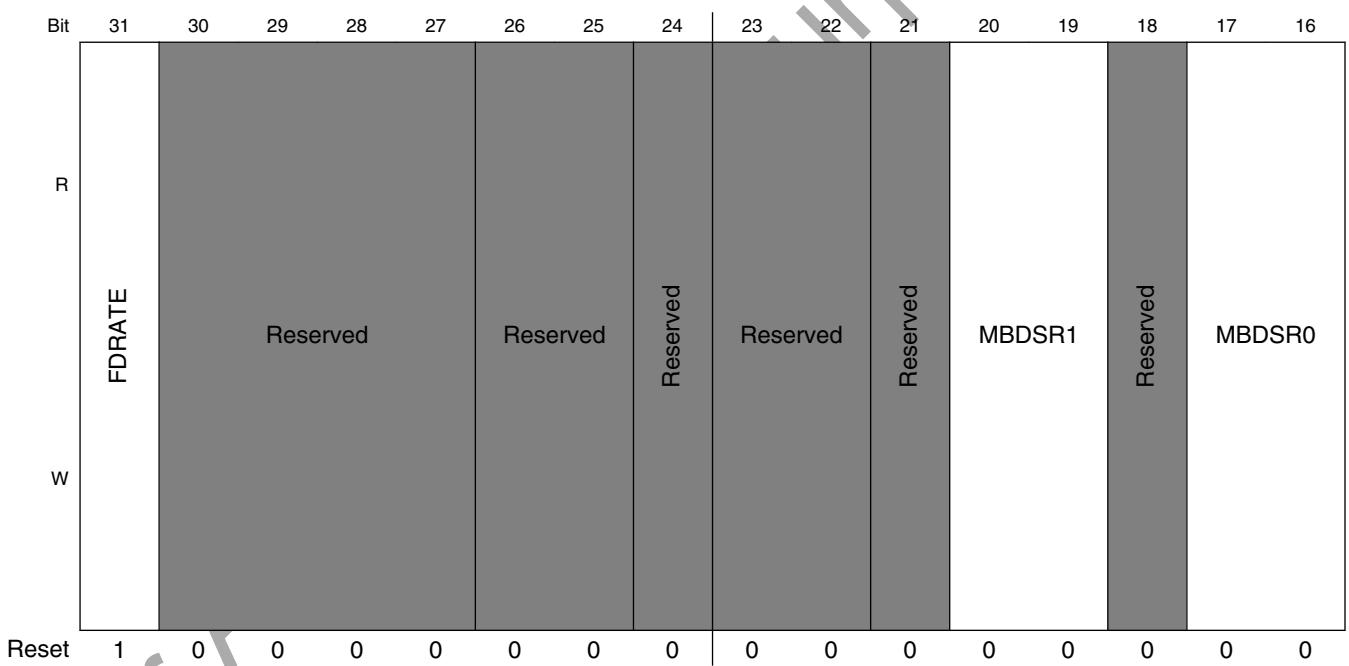
Payload Size	Maximum number of Message Buffers per RAM block
8 bytes	32
16 bytes	21
32 bytes	12
64 bytes	7

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.

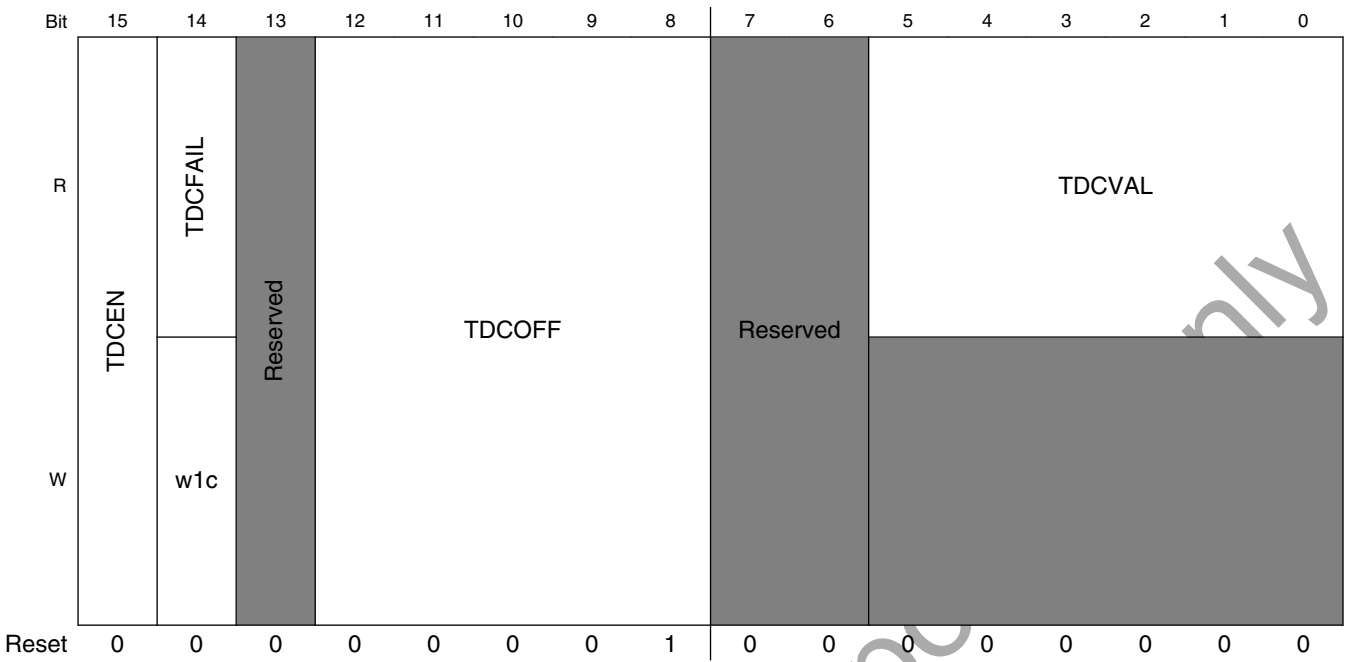
The contents of this register are not affected by soft reset.

Address: 0h base + C00h offset = C00h



For Academic Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required



CAN_FDCTRL field descriptions

Field	Description
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. 0 Transmit a frame in nominal rate. The BRS bit in the Tx MB has no effect. 1 Transmit a frame with bit rate switching if the BRS bit in the Tx MB is recessive.
30–27 Reserved	This field is reserved.
26–25 Reserved	This field is reserved.
24 Reserved	This field is reserved.
23–22 Reserved	This field is reserved.
21 Reserved	This field is reserved.
20–19 MBDSR1	Message Buffer Data Size for Region 1 This two bit field selects the data size (8, 16, 32 or 64 bytes) for the region R1 of Message Buffers allocated in RAM. It can be written in Freeze Mode only. 00 Selects 8 bytes per Message Buffer.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_FDCTRL field descriptions (continued)

Field	Description
	01 Selects 16 bytes per Message Buffer. 10 Selects 32 bytes per Message Buffer. 11 Selects 64 bytes per Message Buffer.
18 Reserved	This field is reserved.
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. 00 Selects 8 bytes per Message Buffer. 01 Selects 16 bytes per Message Buffer. 10 Selects 32 bytes per Message Buffer. 11 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:** TDC must be disabled when the Loop Back Mode is enabled (see CAN_CTRL1[LPB] register). 0 TDC is disabled 1 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. 0 Measured loop delay is in range. 1 Measured loop delay is out of range.
13 Reserved	This field is 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	This field is reserved.
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).

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_FDCTRL field descriptions (continued)

Field	Description
	See Protocol timing for more details on how the loop delay measurement is performed.

46.4.34 CAN FD Bit Timing Register (CAN_FDCBT)

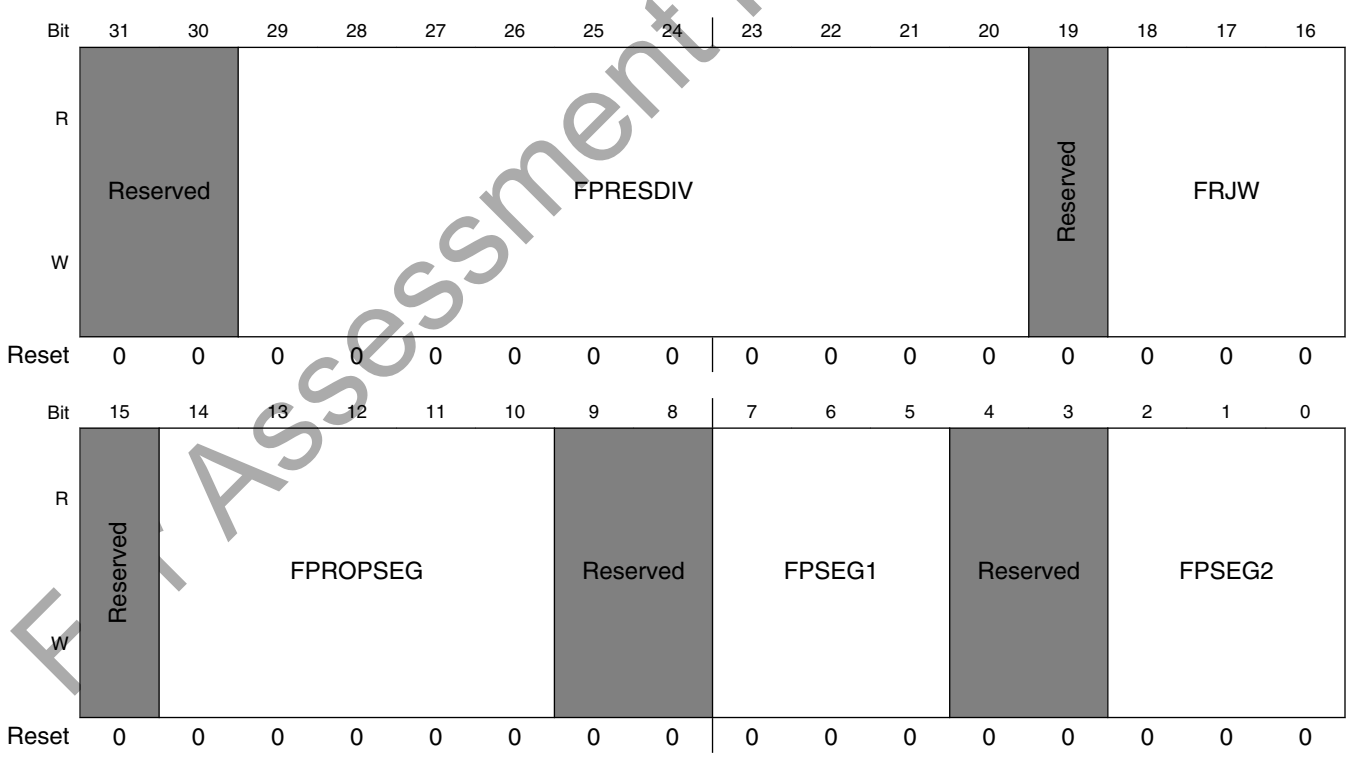
This register stores the CAN bit timing variables used in the data phase of CAN FD messages when the CAN_FDCTRL[FDRATE] is set, compatible with CAN FD specification. FPRESDIV, FPROPSEG, FPSEG1, FPSEG2 and FRJW are used to define the time quantum duration, the number of time quanta per CAN bit and the 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.

Address: 0h base + C04h offset = C04h



For Assessment Purposes Only

CAN_FDCBT field descriptions

Field	Description
31–30 Reserved	This field is 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	This field is 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	This field is 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	This field is 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	This field is reserved.
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. Phase Segment 2 = (FPSEG2 + 1) × Time-Quanta.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

CAN_FDCBT field descriptions (continued)

Field	Description
	Time-Quantum = one Sclock period.

46.4.35 CAN FD CRC Register (CAN_FDCRC)

This register provides information about the CRC of transmitted messages.

FlexCAN uses different CRC polynomials for different frame formats, as shown below.

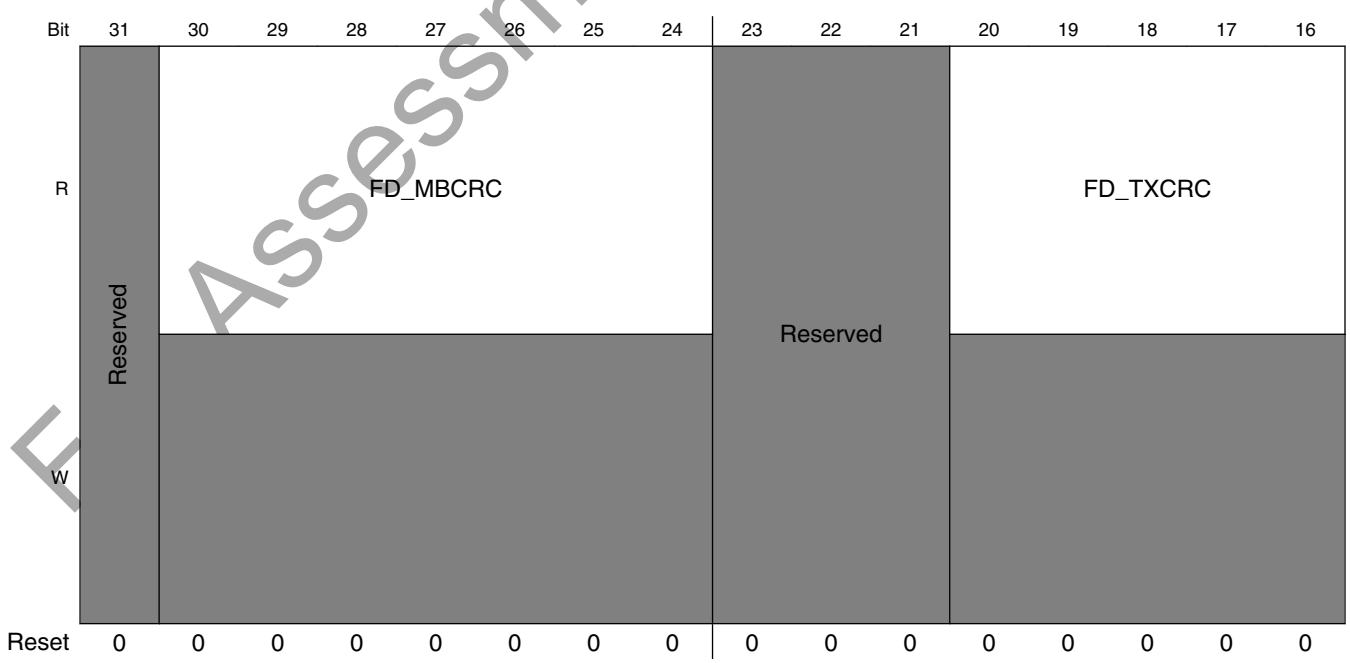
The CRC_15 polynomial is used for all frames in CAN format. The CRC_17 polynomial is used for frames in CAN FD format with a DATA FIELD up to sixteen bytes. The CRC_21 polynomial is used for frames in CAN FD format with a DATA FIELD longer than sixteen bytes. Each polynomial shown below results in a 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: $(x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^6 + x^4 + x^3 + x^1 + 1)$

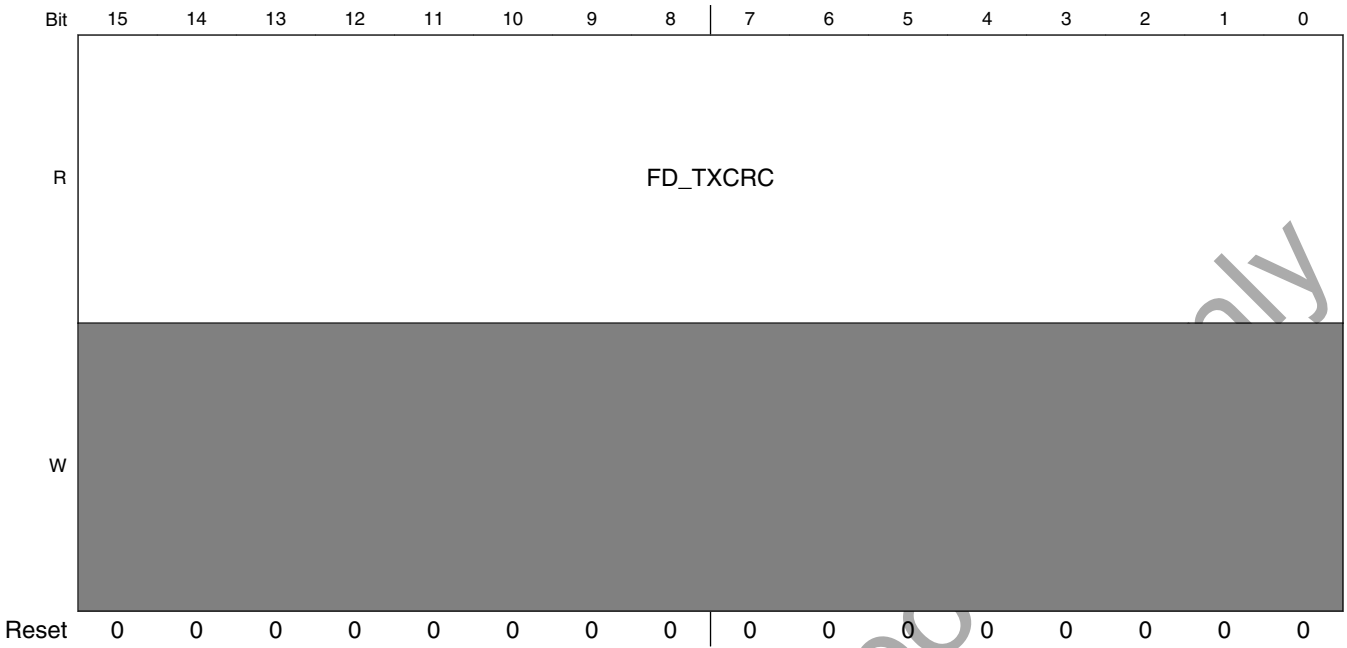
CRC_21 = 0x302899: $(x^{21} + x^{20} + x^{13} + x^{11} + x^7 + x^4 + x^3 + 1)$

Address: 0h base + C08h offset = C08h



For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required



CAN_FDCRC field descriptions

Field	Description
31 Reserved	This field is 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	This field is reserved.
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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

46.4.36 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 46-9. 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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 46-10 and Table 46-11. See Functional description for additional information.

Only

Table 46-10. 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

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-10. 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
					OVERRUN. See Matching process 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.

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-10. 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
					See Matching process for details.
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 46-11. 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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-11. Message buffer code for Tx buffers

CODE Description	Tx Code BEFORE tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
				transmitted unconditionally once, 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 46-10, Table 46-11, 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.

For Assessment Purpo

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 46-9) . 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 46-12).

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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-12. 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

46.4.37 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 accomodate a number of Message Buffers which depends on the configuration provided by CAN_FDCTRL[MBDSRn] bit fields as shown in table below.

Table 46-13. 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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 46-14. 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

46.4.38 Rx FIFO structure

When the CAN_MCR[RFEN] bit is set, the memory area from 0x80 to 0xDC (which is normally occupied by MBs 0–5) is used by the reception FIFO engine.

The region 0x80-0x8C contains the output of the FIFO which must be read by the CPU as a message buffer. This output contains the oldest message that has been received but not yet read. The region 0x90-0xDC is reserved for internal use of the FIFO engine.

An additional memory area, which starts at 0xE0 and may extend up to 0x2DC (normally occupied by MBs 6–37) depending on the CAN_CTRL2[RFFN] field setting, contains the ID filter table (configurable from 8 to 128 table elements) that specifies filtering criteria for accepting frames into the FIFO.

Out of reset, the ID filter table flexible memory area defaults to 0xE0 and extends only to 0xFC, which corresponds to MBs 6 to 7 for RFFN = 0, for backward compatibility with previous versions of FlexCAN.

The following shows the Rx FIFO data structure.

Table 46-15. Rx FIFO structure

	31 28	24 23	22	21	20	19 18 17 16	15 8	7 0
0x80	IDHIT		SRR	IDE	RTR	DLC	TIME STAMP	
0x84		ID standard				ID extended		
0x88	Data byte 0		Data byte 1				Data byte 2	Data byte 3
0x8C	Data byte 4		Data byte 5				Data byte 6	Data byte 7
0x90 to 0xDC	Reserved							
0xE0	ID filter table element 0							

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-15. Rx FIFO structure (continued)

0xE4	ID filter table element 1
0xE8 to 0x2D4	ID filter table elements 2 to 125
0x2D8	ID filter table element 126
0x2DC	ID filter table element 127
	= Unimplemented or reserved

Each ID filter table element occupies an entire 32-bit word and can be compounded by one, two, or four Identifier Acceptance Filters (IDAF) depending on the CAN_MCR[IDAM] field setting. The following figures show the IDAF indexation.

The following table shows the three different formats of the ID table elements. Note that all elements of the table must have the same format. See Rx FIFO for more information.

Table 46-16. ID table structure

Format	31	30	29 ... 24	23 ... 16	15	14	13 ... 8	7 ... 1	0
A	RTR	IDE	RXIDA (standard = 29–19, extended = 29–1)						
B	RTR	IDE	RXIDB_0 (standard = 29–19, extended = 29–16)		RTR	IDE	RXIDB_1 (standard = 13–3, extended = 13–0)		
C	RXIDC_0 (std/ext = 31–24)			RXIDC_1 (std/ext = 23–16)	RXIDC_2 (std/ext = 15–8)			RXIDC_3 (std/ext = 7–0)	
	= Unimplemented or Reserved								

RTR — Remote Frame

This bit specifies if Remote Frames are accepted into the FIFO if they match the target ID.

1 = Remote Frames can be accepted and data frames are rejected

0 = Remote Frames are rejected and data frames can be accepted

IDE — Extended Frame

Specifies whether extended or standard frames are accepted into the FIFO if they match the target ID.

1 = Extended frames can be accepted and standard frames are rejected

0 = Extended frames are rejected and standard frames can be accepted

Freescale Confidential Proprietary - Non-Disclosure Agreement required

RXIDA — Rx Frame Identifier (Format A)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, only the 11 most significant bits (29 to 19) are used for frame identification. In the extended frame format, all bits are used.

RXIDB_0, RXIDB_1 — Rx Frame Identifier (Format B)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, the 11 most significant bits (a full standard ID) (29 to 19 and 13 to 3) are used for frame identification. In the extended frame format, all 14 bits of the field are compared to the 14 most significant bits of the received ID.

RXIDC_0, RXIDC_1, RXIDC_2, RXIDC_3 — Rx Frame Identifier (Format C)

Specifies an ID to be used as acceptance criteria for the FIFO. In both standard and extended frame formats, all 8 bits of the field are compared to the 8 most significant bits of the received ID.

IDHIT — Identifier Acceptance Filter Hit Indicator

This 9-bit field indicates which Identifier Acceptance Filter was hit by the received message that is in the output of the Rx FIFO. See Rx FIFO for more information.

46.5 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). The memory corresponding to the first 38 MBs can be configured to support a FIFO reception scheme with a powerful ID filtering mechanism, capable of checking incoming frames against a table of IDs (up to 128 extended IDs or 256 standard IDs or 512 8-bit ID slices), with individual mask register for up to 32 ID Filter Table elements.

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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 46-10). Similarly, a Tx MB with a 0b1000 or 0b1001 code is also inactive (refer to Table 46-11).

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.

46.5.1 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 abortion of the transmission. 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. Then read back the CODE field to check if the transmission was aborted or transmitted (see Transmission abort mechanism). If backwards compatibility is desired (CAN_MCR[AEN] bit is negated), just write the INACTIVE code (0b1000) to the CODE field to inactivate the MB but then the pending frame may be transmitted without notification (see Mailbox inactivation).
3. Write the ID word.
4. Write the data bytes.
5. Write the DLC, Control, and CODE fields of the Control and Status word to activate the MB. When CAN_MCR[FDEN] is set, write also the EDL, BRS and ESI bits.

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

For Assessment Purposes Only

Interrupt Flag Register and 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 46-10 and Table 46-11 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 CPU. This means that the CPU must clear the corresponding IFLAG bit before starting to prepare this MB for a new transmission or reception.

46.5.2 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*.

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.

For Assessment Purposes Only

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.

46.5.2.1 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.

46.5.2.2 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.

46.5.2.2.1 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 46-17. 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)

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

46.5.2.2.2 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 46-18. 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.

46.5.2.3 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.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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.

For Assessment Purposes Only

46.5.3 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 46-10 and Table 46-11 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.

For Assessment Purposes Only

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 46-10 . 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.

To be able to receive CAN frames through the Rx FIFO, the CPU must enable and configure the Rx FIFO during Freeze mode (see Rx FIFO). Upon receiving the Frames Available in Rx FIFO interrupt (see the description of the BUF5I bit "Frames available in Rx FIFO" bit in the CAN_IFLAG1 register), the CPU should service the received frame using the following procedure:

1. Read the Control and Status word (optional: needed only if a mask was used for IDE and RTR bits)
2. Read the ID field (optional: needed only if a mask was used)
3. Read the Data field
4. Read the CAN_RXFIR register (optional)
5. Clear the Frames Available in Rx FIFO interrupt by writing 1 to CAN_IFLAG1[BUF5I] bit (mandatory: releases the MB and allows the CPU to read the next Rx FIFO entry)

For Assessment Purposes Only

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. Read the last Data Bytes (read 0x8C address is mandatory)

46.5.4 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. If the FIFO is enabled, the priority of scanning can be selected between Mailboxes and FIFO filters. The matching starts from the lowest number Message Buffer toward the higher ones. If no match is found within the first structure then the other is scanned subsequently. In the event that the FIFO is full, the matching algorithm always looks for a matching MB outside the FIFO region.

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 the FIFO table or in one of the Mailboxes, the contents of the Rx SMB are transferred to the FIFO or to the matched Mailbox by the move-in process. If any CAN protocol error is detected then no match results are transferred to the FIFO or to the matched Mailbox at the end of reception.

The matching process scans all matching elements of both Rx FIFO (if enabled) and 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

For Assessment Purposes Only

frame on the CAN bus. The Rx SMB has the same structure of a Mailbox. The reception structures (Rx FIFO or 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 46-19. 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
FIFO[5]	-	-	-	cmp_msk	cmp_msk	cmp_msk	-

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.
5. SMB[IDE] and SMB[RTR] are not taken into account when IDAM is type C.

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 Rx FIFO is not full

The scan order for Mailboxes and Rx FIFO 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:

For Assessment

1. the first free-to-receive matched Mailbox;
2. the last non free-to-receive matched Mailbox.

It is possible to select the priority of scan between Mailboxes and Rx FIFO by the CAN_CTRL2[MRP] bit.

If the selected priority is Rx FIFO first:

- If the Rx FIFO is a matched structure and is free-to-receive, then the Rx FIFO is the matching winner regardless of the scan for Mailboxes
- Otherwise (the Rx FIFO is not a matched structure or is not free-to-receive), then the matching winner is searched among Mailboxes as described above

If the selected priority is Mailboxes first:

- If a free-to-receive matched Mailbox is found, it is the matching winner regardless of the scan for Rx FIFO
- If no matched Mailbox is found, then the matching winner is searched in the scan for the Rx FIFO
- If both conditions above are not satisfied and a non free-to-receive matched Mailbox is found, then the matching winner determination is conditioned by the CAN_MCR[IRMQ] bit:
- - If CAN_MCR[IRMQ] bit is negated, the matching winner is the first matched Mailbox
  - If CAN_MCR[IRMQ] bit is asserted, the matching winner is the Rx FIFO if it is a free-to-receive matched structure; otherwise, the matching winner is the last non free-to-receive matched Mailbox

See the following table for a summary of matching possibilities.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-20. Matching possibilities and resulting reception structures

RFEN	IRMQ	MRP	Matched in MB	Matched in FIFO	Reception structure	Description
No FIFO, only MB, match is always MB first						
0	0	X[1]	None[2]	-[3]	None	Frame lost by no match
0	0	X	Free[4]	-	FirstMB	
0	1	X	None	-	None	Frame lost by no match
0	1	X	Free	-	FirstMb	
0	1	X	NotFree	-	LastMB	Overrun
FIFO enabled, no match in FIFO is as if FIFO does not exist						
1	0	X	None	None[5]	None	Frame lost by no match
1	0	X	Free	None	FirstMB	

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 46-20. Matching possibilities and resulting reception structures (continued)

RFEN	IRMQ	MRP	Matched in MB	Matched in FIFO	Reception structure	Description
1	1	X	None	None	None	Frame lost by no match
1	1	X	Free	None	FirstMb	
1	1	X	NotFree	None	LastMB	Overrun
FIFO enabled, Queue disabled						
1	0	0	X	NotFull[6]	FIFO	
1	0	0	None	Full[7]	None	Frame lost by FIFO full (FIFO Overflow)
1	0	0	Free	Full	FirstMB	
1	0	0	NotFree	Full	FirstMB	
1	0	1	None	NotFull	FIFO	
1	0	1	None	Full	None	Frame lost by FIFO full (FIFO Overflow)
1	0	1	Free	X	FirstMB	
1	0	1	NotFree	X	FirtsMb	Overrun
FIFO enabled, Queue enabled						
1	1	0	X	NotFull	FIFO	
1	1	0	None	Full	None	Frame lost by FIFO full (FIFO Overflow)
1	1	0	Free	Full	FirstMB	
1	1	0	NotFree	Full	LastMb	Overrun
1	1	1	None	NotFull	FIFO	
1	1	1	Free	X	FirstMB	
1	1	1	NotFree	NotFull	FIFO	
1	1	1	NotFree	Full	LastMb	Overrun

1. This is a don't care condition.
2. Matched in MB "None" means that the frame has not matched any MB (free-to-receive or non-free-to-receive).
3. This is a forbidden condition.
4. Matched in MB "Free" means that the frame matched at least one MB free-to-receive regardless of whether it has matched MBs non-free-to-receive.
5. Matched in FIFO "None" means that the frame has not matched any filter in FIFO. It is as if the FIFO didn't exist (CAN_CTRL2[RFEN]=0).
6. Matched in FIFO "NotFull" means that the frame has matched a FIFO filter and has empty slots to receive it.
7. Matched in FIFO "Full" means that the frame has matched a FIFO filter but couldn't store it because it has no empty slots to receive it.

FC Asse

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 the FIFO is disabled, 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 (in addition to the full featured FIFO) 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_RXFGMASK, 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.

46.5.5 Receive Process under Pretended Networking Mode

Pretended Networking mode adds specific wake up functionality in low power mode (Stop mode). When Pretended Network mode (PN) is enabled by asserting the PNET_EN bit in the Module Configuration Register (CAN_MCR)), FlexCAN continues processing Rx CAN messages under low power mode, able to detect specific wake up messages by

For Assessment Purposes Only

filtering them against ID and payload target values using a pre-selected matching criteria. Wake up functionality is not available for messages in CAN FD format. While in Pretended Networking mode, CAN FD format messages are ignored.

PN registers are located in the 0x0B00 - 0x0B7C address range and can be written only in Freeze mode. These registers are used for writing PN configuration (both control and target values) prior entering to Pretended Networking mode, and for reading wake up flags and the received message ID and data when returning back to normal mode after wake up. The CPU must wait for CAN_MCR[LPMACK] to be negated before performing any access to FlexCAN PN registers.

PN control registers are described in Pretended Networking Control 1 Register (CAN_CTRL1_PN) and Pretended Networking Control 2 Register (CAN_CTRL2_PN). The control bit fields that configure the filtering criteria are:

- PLFS: payload filtering selection.
- IDFS: ID filtering selection.
- FCS: filtering combination selection.

PN target values are:

- FLT_IDE: IDE target value used to filter the incoming message by its format (standard or extended).
- FLT_RTR: RTR target value used to filter the incoming message by its type (data or remote frame).
- FLT_DLC_HI and FLT_DLC_LO: target DLC range used to filter the size of payload part of an incoming message.
- FLT_ID1: ID target value used to filter the incoming message ID (equal to, smaller than or equal, greater than or equal, or the lower limit value in an ID range).
- FLT_ID2: ID target value used as the upper limit in an ID range.
- PL1: Payload target value used to filter the incoming message payload (equal to, smaller than or equal, greater than or equal, or the lower limit value in a payload range).
- PL2: Payload target value used as the upper limit in a payload range.

IDE, RTR, ID and payload filters have their respective masks. These masks determine which bits are taken into account in equality comparisons ("1's" in certain mask positions) and which ones are don't care ("0's" in other mask positions). ID and payload masks are used only for exact ID and/or exact Payload comparisons.

The ID of Rx incoming messages can be filtered based on the following criteria:

- A match with the exact ID value by detecting the equality between the ID field of the incoming message and the content of target CAN_FLT_ID1 register. The ID mask is used.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- A match with the maximum range of ID, i.e. any message with ID value smaller than or equal to the content of target CAN_FLT_ID1 register is accepted. The ID mask is not used.
- A match with the minimum range of ID, i.e. any message with ID value greater than or equal to the content of target CAN_FLT_ID1 register is accepted. The ID mask is not used.
- A match inside a range of IDs, i.e. any message with an ID value that is greater than or equal to the content of target CAN_FLT_ID1 register and smaller than or equal to the content of target CAN_FLT_ID2_IDMASK register is accepted. The ID mask is not used.

See CAN_CTRL1_PN[IDFS] in Pretended Networking Control 1 Register (CAN_CTRL1_PN).

The above criteria for ID filtering must be coherent with FLT_IDE and FLT_RTR target values in CAN_FLT_ID1 register. Only Rx frames that match the respective IDE and RTR bits to the contents of FLT_IDE and FLT_RTR bit fields will be compared. When range of IDs is selected (CAN_CTRL1_PN[IDFS] = 11), both FLT_ID1 and FLT_ID2 are referred to the same FLT_IDE and FLT_RTR bits in CAN_FLT_ID1 register.

The ID mask is applied only to the exact ID comparison filtering option (CAN_CTRL1_PN[IDFS] = 00) to determine which bits are taken into account in the comparison. For the exact match option, the mask can select any bit within the ID field. For maximum range, minimum range and inside range comparisons, the ID mask is not considered.

The IDE and RTR masks are applied in both exact and range ID comparison filtering options to determine which bits are taken into account in comparison.

Similarly to the ID criteria, 64-bit data or payloads (PL) of Rx incoming messages can be filtered based on the following criteria:

- A match with the exact payload value by detecting the equality between the payload field of the incoming message and the content of PL1 register. The payload mask is used.
- A match with the maximum range of payload, i.e. any message with payload value smaller than or equal to the content of PL1 register is accepted. The payload mask is not used.
- A match with the minimum range of payload, i.e. any message with payload value greater than or equal to the content of PL1 register is accepted. The payload mask is not used.
- A match inside a range of payloads, i.e. any message with a payload value that is greater than or equal to the content of PL1 register and smaller than or equal to the content of PL2 register is accepted. The payload mask is not used.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

See CAN_CTRL1_PN[PLFS] in Pretended Networking Control 1 Register (CAN_CTRL1_PN).

The above criteria for payload filtering must be coherent with FLT_DLC upper and lower limit values in CAN_FLT_DLC register. The payload of a Rx incoming message is filtered in accordance to the selected criteria only if the DLC value of the Rx incoming message is inside a DLC range:

- greater than or equal to the FLT_DLC_LO (lower limit) and
- lower than or equal to the FLT_DLC_HI (upper limit).

Conversely, a DLC value out of the specified range results in mismatch. By making FLT_DLC_LO = FLT_DLC_HI, only payloads of specified quantity of bytes will be filtered. DLC is not maskable.

When the inside range of payloads option is selected (CAN_CTRL1_PN[PLFS] = 11), both PL1 and PL2 are considered with the 8-byte data length. All the data bytes excluded by the DLC of the received message are considered with value zero.

Payload mask is only used in the exact match option (CAN_CTRL1_PN[PLFS] = 00) to select which bits or bytes in the 8-byte data field of both Rx incoming message and the contents of PL1 register are selected for matching. Mask length must be in accordance to the expected range of DLC values. For maximum range, minimum range and inside range comparisons, the payload mask is not considered.

When a remote frame is received by FlexCAN and the CAN_CTRL1[FCS] bit is configured to select the payload comparison, the payload filtering is not considered and the comparison results in a mismatch.

Rx incoming messages can also be filtered based upon the quantity and rate of message reception, specifically:

- Several messages that match the filtering criteria for ID or payload a predefined quantity of times. This quantity can be configured in the 1 to 255 range. See the Pretended Networking Control 1 Register (CAN_CTRL1_PN).
- No message matching the filtering criteria for ID or payload up to a timeout trigger. That is, non-reception of a matching message for a defined quantity of time. See the Pretended Networking Control 2 Register (CAN_CTRL2_PN).

FlexCAN can generate a wake up timeout event from an internal timer with associated comparator circuitry capable to generate a timeout flag when the counting reaches the pre-defined timeout value, as specified in CAN_CTRL2_PN[MATCHTO].

The above filtering criteria can be used together as follows:

- Message ID filtering only.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- Message ID filtering and Payload filtering.
- Message ID filtering only occurring N times.
- Message ID filtering and Payload filtering occurring N times.

The timeout counter runs concurrently with the reception filtering process. Both engines, timeout counter and message filtering, are independent. If an incoming message matches the selected filter criteria, the timeout counter keeps counting until the CPU wakes-up. Conversely, if the timeout counter reaches the target value then the message filtering process continue to filter incoming messages until the CPU wakes-up. The CAN_WU_MTC[MCOUNTER] field will report the number of matched messages occurred under Pretended Networking mode up to the moment the CPU wakes up.

Under Pretended Networking mode, the wake up event that may occur will set the respective wake up flag (see Pretended Networking Wake Up Match Register (CAN_WU_MTC)):

- In case of a successful matching in accordance to the selected filtering criteria, the CAN_WU_MTC[WUMF].
- In case of a timeout trigger, the CAN_WU_MTC[WTOF].

Any of these flags will generate interrupts to the CPU, provided the respective mask bits are enabled (WUMF_MSK or WTOF_MSK in CAN_CTRL1_PN register).

There are four WMB's (Wake up Message Buffers) used to store incoming messages in Pretended Networking mode. Up to four messages can be stored (see Pretended Networking Wake Up Message Buffer Registers (CAN_WMB0 - CAN_WMB3)). When CTRL1_PN[NMATCH] value is 1, just one message is received if matching the filtering criteria, and this message is stored in CAN_WMB0. If NMATCH value is between 2 and 4, CAN_WMB1, CAN_WMB2 and CAN_WMB3 are used to store the second,third and fourth matching messages, respectively. If NMATCH is greater than 4, the last four matching messages are stored in the WMBs, respecting the WMB index to indicate the arrival order, the latest is stored in CAN_WMB3. Only the valid data bytes of the incoming match message is stored in data field of WMBs. The non-valid data bytes is read as zero. In case of DLC=0 and RTR=1 the data field is filled with zero. In any of the above cases, the wake up interrupt is generated just when the filtering criteria is completed and CAN_CTRL1_PN[WUMF_MSK] is enabled.

When a non-match wake-up event occurs (timeout or external) and MCOUNTER register is equal or greater than 4, the message stored in WMB0 does not have a valid content. The CAN_WMB0 is used as buffer for the current message in CAN bus. Messages received during Pretended Networkig mode don't have timestamps and respective field in the WMB structure must be ignored.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Under low power mode(Stop), all processes are shut down except for the PN functionality inside CAN_PE sub-block, that is kept clocked by the Oscillator clock (see Clock domains and restrictions). FlexCAN continues to receive Rx incoming messages and just compares them against the predefined target values and in accordance to the selected filtering criteria. The matching, arbitration, move-in and move-out processes, normally available in Normal mode, are not performed under Pretended Networking mode.

The FlexCAN under Pretended Networking reacts to messages on the CAN bus in the same manner as in Normal mode (i.e. generates acknowledge bits, detect and count errors, etc.).

46.5.6 Move process

There are two types of move process: move-in and move-out.

46.5.6.1 Move-in

The move-in process is the copy of a message received by an Rx SMB to a Rx Mailbox or FIFO that has matched it. If the move destination is the Rx FIFO, attributes of the message are also copied to the CAN_RXFIR FIFO. 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:

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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. Push IDHIT into the RXFIR FIFO if the message is destined to the Rx FIFO.
2. Read all data words from the Rx SMB in accordance to the selected payload size for the Rx storage element.
3. 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.
4. Read the Control/Status and ID words from the Rx SMB.
5. 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 exception is if the move-in destination is an Rx FIFO Message Buffer, then the process cannot be cancelled.

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.

46.5.6.2 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

For Assessment Purposes Only

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.

46.5.7 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.

46.5.7.1 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

The abort procedure can be summarized as follows:

- CPU checks the corresponding IFLAG and clears it, if asserted.
- CPU writes 0b1001 into the CODE field of the C/S word.
- CPU waits for the corresponding IFLAG indicating that the frame was either transmitted or aborted.
- CPU reads the CODE field to check if the frame was either transmitted (CODE=0b1000) or aborted (CODE=0b1001).
- It is necessary to clear the corresponding IFLAG in order to allow the MB to be reconfigured.

46.5.7.2 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:

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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).

NOTE

Message Buffers that are part of the Rx FIFO cannot be inactivated. There is no write protection on the FIFO region by FlexCAN. CPU must maintain data coherency in the FIFO region when RFEN is asserted.

46.5.7.3 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 are not part of the FIFO and have a code different than INACTIVE (0b0000) or EMPTY[1] (0b0100). Also, Tx MBs can not be locked.

Suppose, for example, that the FIFO is disabled and 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

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.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

arrives and the matching algorithm finds out that there are no "free-to-receive" MBs, so it decides to 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.

46.5.8 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.

For Assessment Purposes Only

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)

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

46.5.8.1 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.

For Assessment Purposes Only

NOTE

CAN_RXFIR register contents cannot be read after DMA completes the FIFO read. The IDHIT information is also available in the C/S word at address 0x080 (see Rx FIFO structure.

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.

46.5.8.2 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.

46.5.9 CAN protocol related features

This section describes the CAN protocol related features.

46.5.9.1 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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

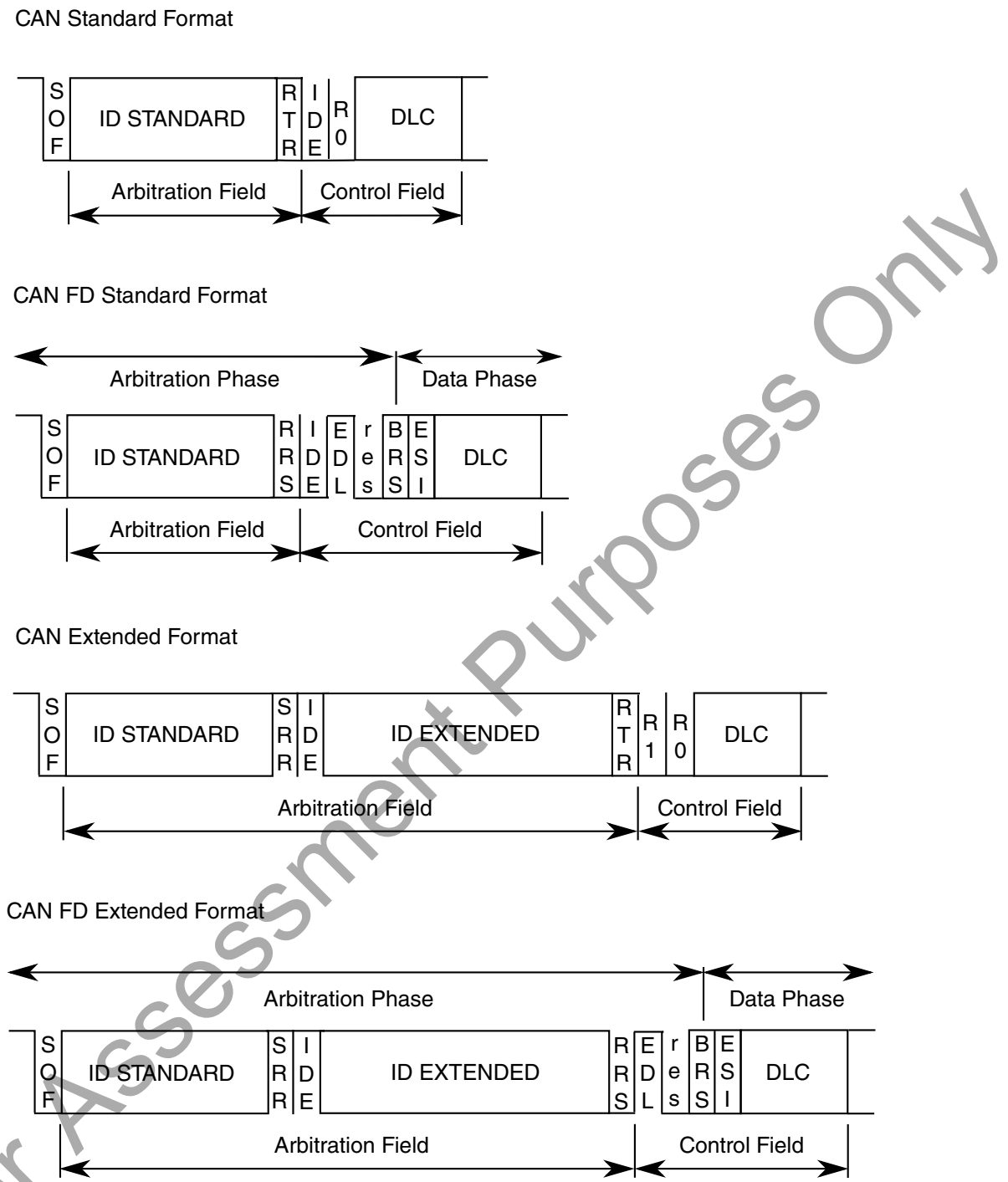


Figure 46-2. CAN message formats

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

For Assessment Purposes Only

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.

For Assessment Purposes Only

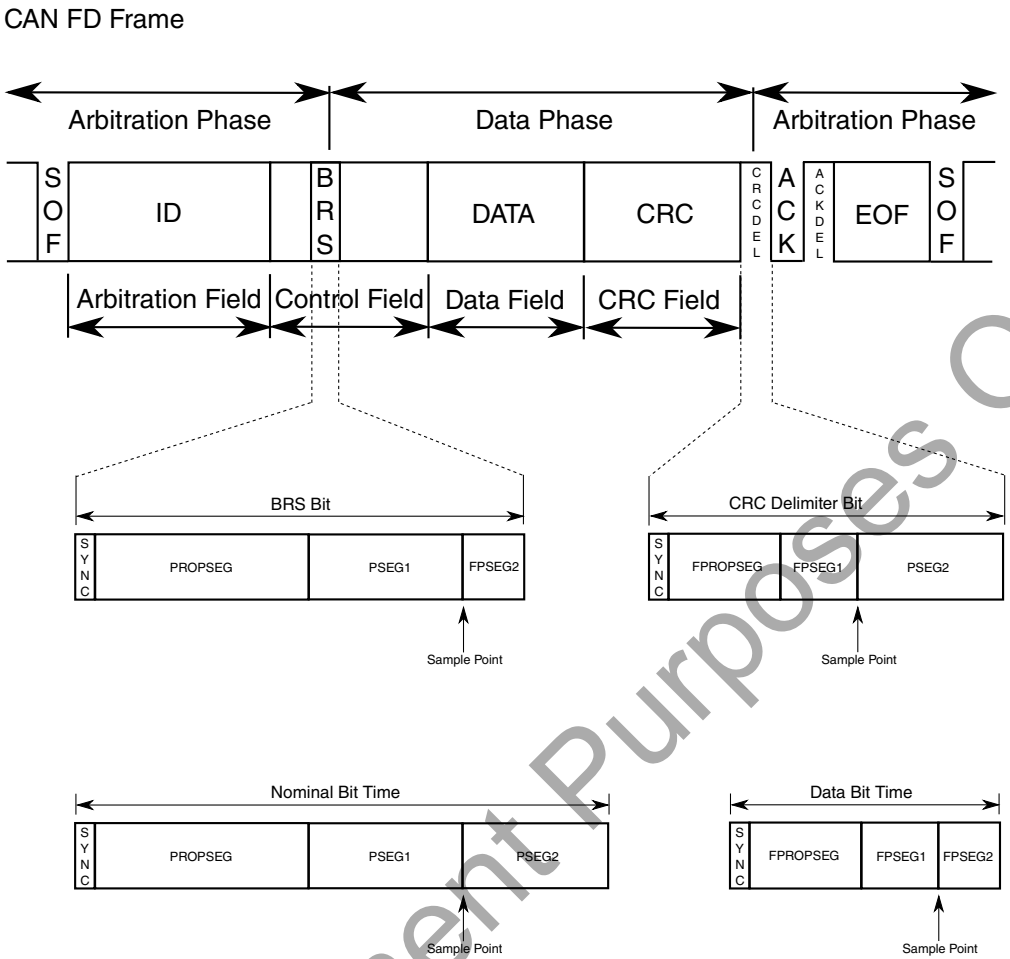


Figure 46-3. Bit rate switching mechanism for CAN FD messages

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 8 time quanta, the bit rate in the data phase is 5 Mbit/s.

For Assessment Purposes Only

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.

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 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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

46.5.9.2 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 transmistted bit and the received bit within the current CAN bit time interval.

FlexCAN supports an optional Transceiver Delay Compensation (TDC) mechanism that 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.

For Assessment Purposes Only

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 15 CANCLK when the delay measurement is too long.

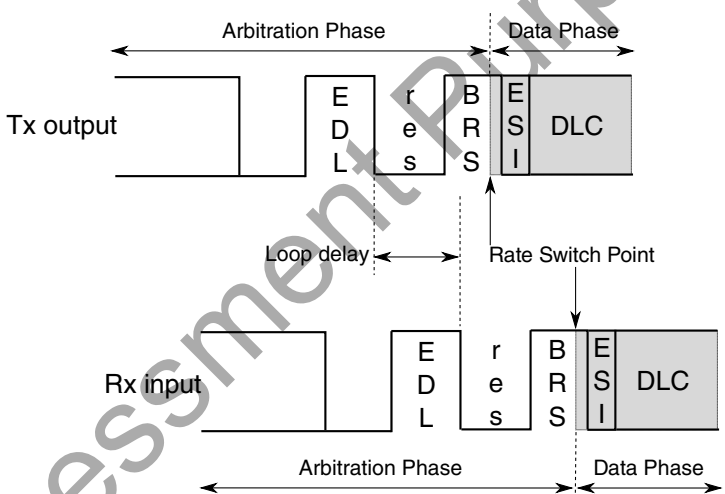


Figure 46-4. Transceiver loop delay measurement

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 (e.g. half of the bit time in the data phase), far away from its edges. Therefore, the TDCOFF value cannot be larger than the CAN bit duration in the data phase.

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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

46.5.9.3 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

For Assessment Purposes Only

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.

46.5.9.4 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

46.5.9.5 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, 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).

For Assessment Purposes Only

46.5.9.6 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).

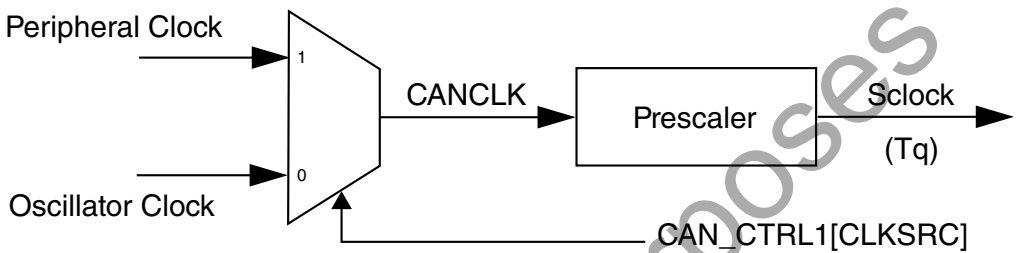


Figure 46-5. CAN engine clocking scheme

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 (CAN_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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

$$Tq = \frac{(PRESDIV + 1)}{f_{CANCLK}}$$

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 46-6, Figure 46-7 and Table 46-21):

- 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 4 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 1 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 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

1. For further explanation of the underlying concepts, see ISO 11898-1. See also the CAN 2.0A/B protocol specification for bit timing.

For Assessment Purposes Only

Information Processing Time (IPT) of 2, which is the value implemented in the FlexCAN module.

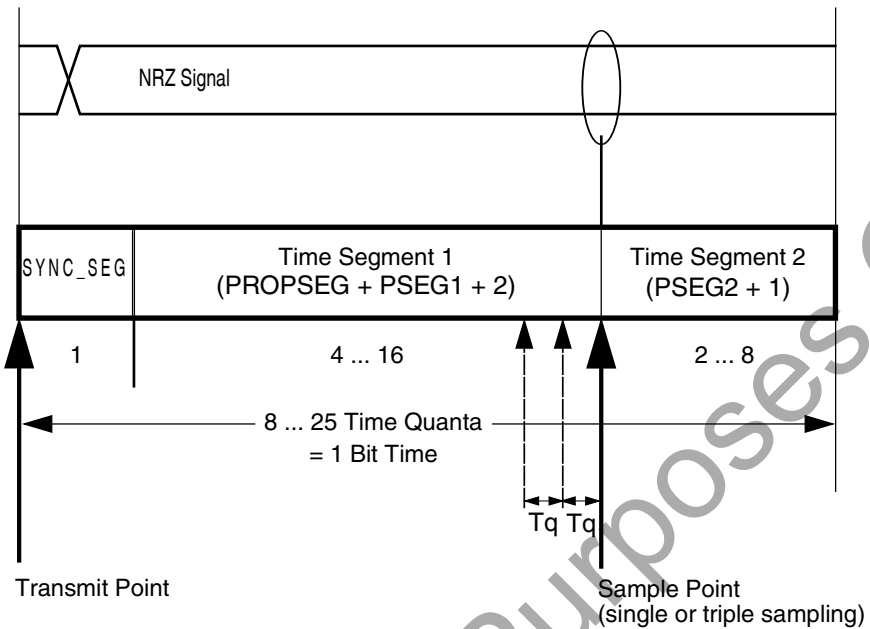


Figure 46-6. Segments within the bit time (example using CAN_CTRL1 bit timing variables for Classical CAN format)

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

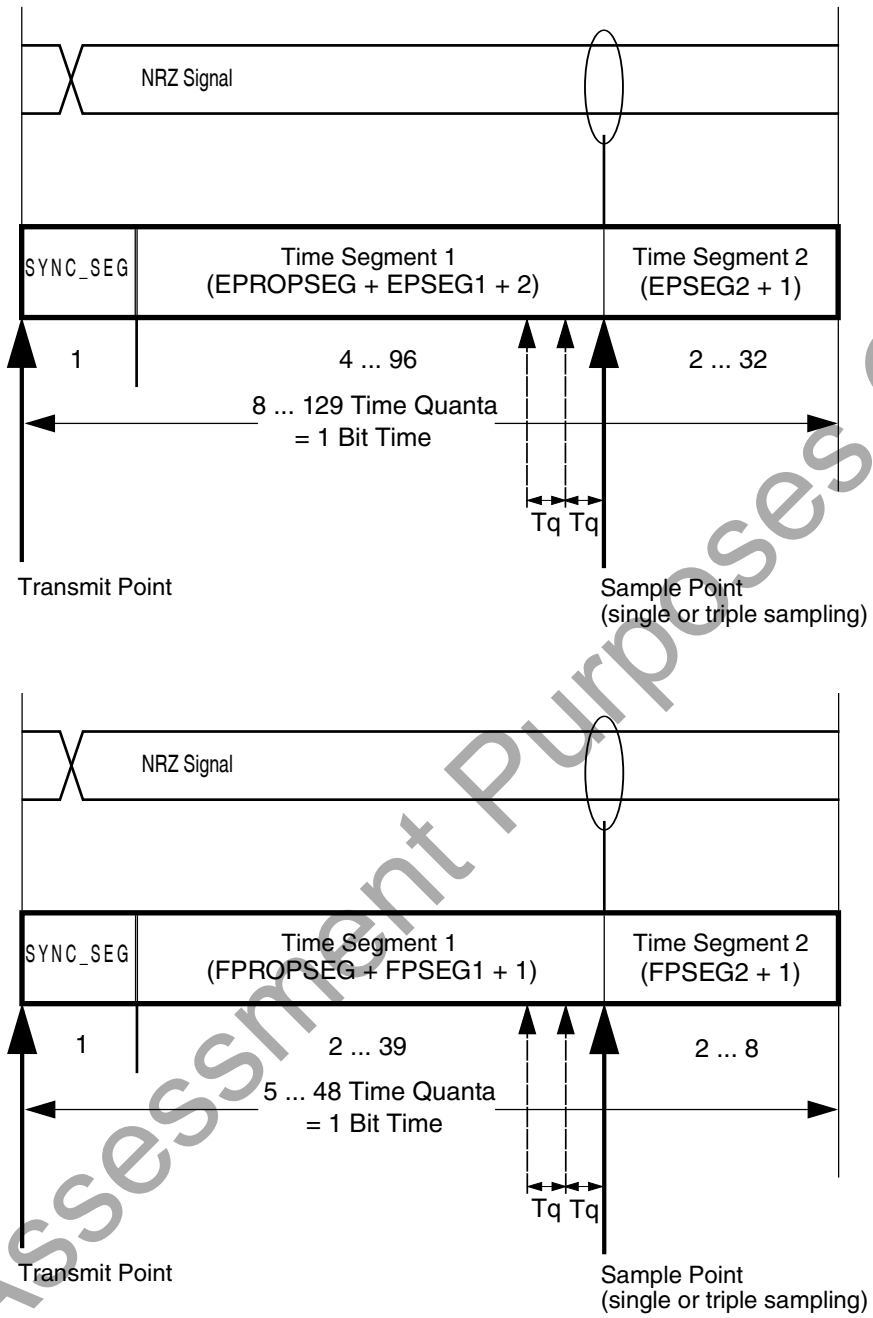


Figure 46-7. Segments within the bit time (example using CAN_CBT and CAN_FDCBT bit timing variables for CAN FD format)

Table 46-21. 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...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Table 46-21. 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 46-22. 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;

For Assessment Purpose Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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.

46.5.9.7 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.

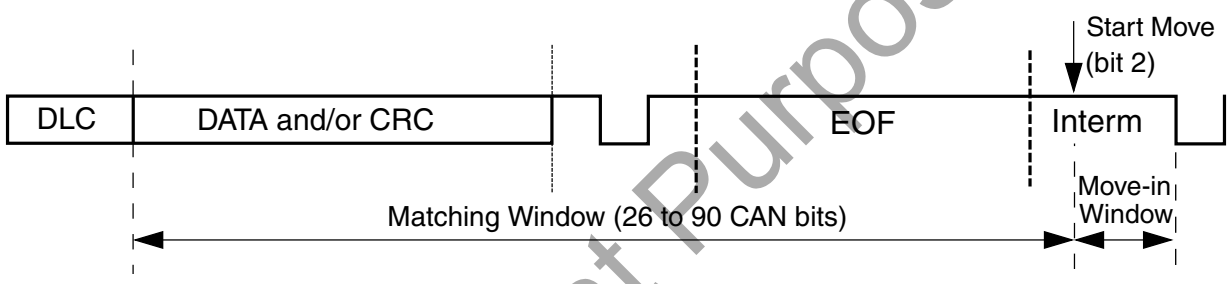


Figure 46-8. Matching and move-in time windows

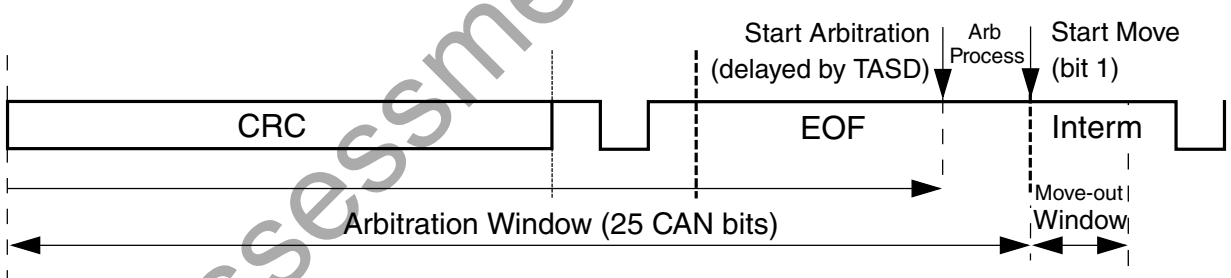


Figure 46-9. Arbitration and move-out time windows

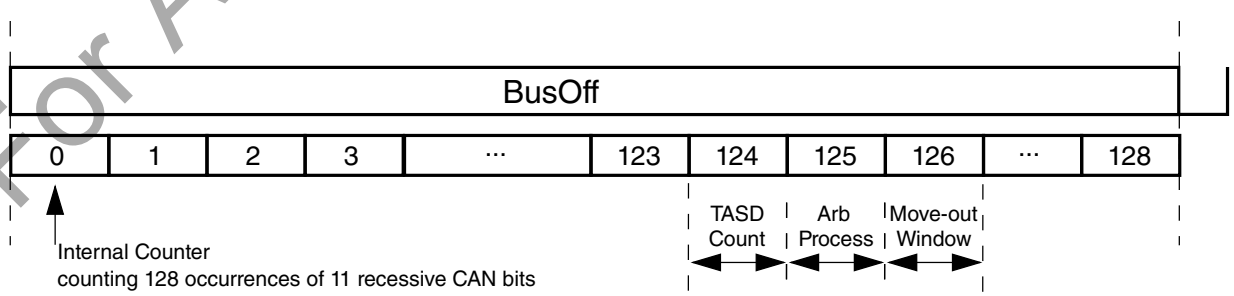


Figure 46-10. Arbitration at the end of bus off and move-out time windows

For Assessment Purposes Only

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.

46.5.9.8 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.

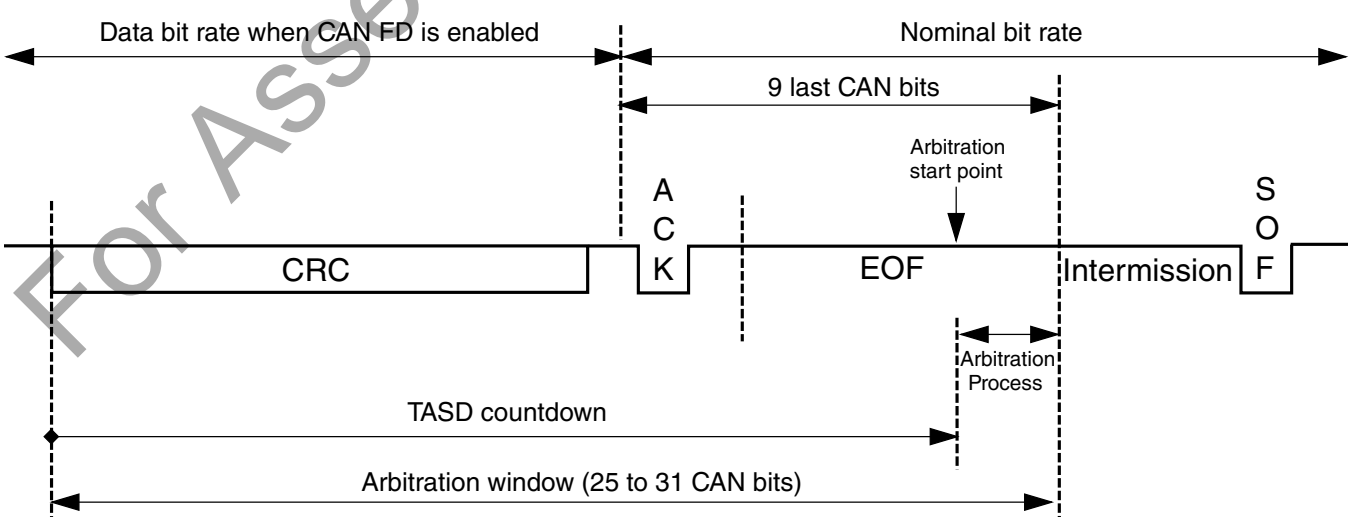


Figure 46-11. Optimal Tx Arbitration start point

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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:

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

$$NMB_{END} = \frac{(9 * CPCB_N) - 4}{2}$$

$$BITRATE_N = \left(\frac{f_{CANCLK}}{[1 + (EPSEG1 + 1) + (EPSEG2 + 1) + (EPROPSEG + 1)] \times (EPRESDIV + 1)} \right)$$

$$BITRATE_F = \left(\frac{f_{CANCLK}}{[1 + (FPSEG1 + 1) + (FPSEG2 + 1) + FPROPSEG] \times (FPRESDIV + 1)} \right)$$

$$CPCB_N = \frac{f_{SYS}}{BITRATE_N}$$

$$CPCB_F = \frac{f_{SYS}}{BITRATE_F}$$

$$CPCB = CPCB_N$$

- 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)

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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 46-23. 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 46-24. 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 46-25. TASD values:

Number of Message Buffers	TASD value	Maximum Bit Rate in Data Phase (Mbaud)
16	24	Invalid
32	23	4.0

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

46.5.10 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 46-26. Minimum number of peripheral clocks per CAN bit for Classical CAN format

Number of Mailboxes	Value of CAN_MCR[RFEN] bit	Minimum number of peripheral clocks per CAN bit
16	0	16
32	0	16
16	1	16
32	1	17

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 46-27. 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

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Assessment Purposes Only

Table 46-27. Number of fast CAN bits in a CAN FD frame (continued)

Minimum number of data bytes	DLC field	NumOfFastBits
8	0x8	85
12	0x9	117
16	0xA	149
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.

$$MinNumClkFastBit_B = 3 \times \left\langle 1 + \frac{f_{SYS}}{f_{CANCLK}} \right\rangle$$

Therefore, the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) is determined by the larger of the two values calculated above.

$$MinNumClkFastBit = Maximum(MinNumClkFastBit_A, MinNumClkFastBit_B)$$

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

$$MinNumClkFastBit_A = \frac{(8.5 \times 96) + 64 - (9 \times 50)}{85} = 5.06$$

$$MinNumClkFastBit_B = 3 \times \left(1 + \frac{50}{40}\right) = 6.75$$

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

$$\text{MinNumClkFastBit} = \text{Maximum}\ (\ 5.06\ ,\ 6.75\ ) = 6.75$$

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 46-28. 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

Freescale Confidential Proprietary - Non-Disclosure Agreement required Review Draft Freescale Semiconductor, Inc. For Internal Use Only

46.5.11 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.

46.5.11.1 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:

For Assessment Purposes Only

- 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.

46.5.11.2 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.

It is not recommended to use Module Disable mode under Pretended Networking mode. Negate the MDIS bit and wait for LPMACK to negate before setting CAN_MCR[PNET_EN].

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 Rx FIFO Global Mask Register. The Rx FIFO Information 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

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

46.5.11.3 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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 46-29. 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.

46.5.11.4 Pretended Networking Mode

This is a special low power mode used to receive wake up messages with low power consumption. This mode can be selected to operate together with Stop mode. Before entering in one of these low power modes, the PNET_EN bit in CAN_MCR Register must be asserted. Once in low power mode, CHI sub-block is shut down and CAN_PE sub-block is kept active, so that the Rx receive process is still active to filter incoming messages (see Receive Process under Pretended Networking Mode) as defined by the configuration registers (see Pretended Networking Control 1 Register (CAN_CTRL1_PN)). Upon detecting a wake up event, a Wake Up interrupt is issued to the system.

To enter in Pretended Networking mode FlexCAN must be in normal mode (neither in Freeze, nor in Disable mode). Under Pretended Networking mode, FlexCAN keeps itself synchronized with the CAN BUS in Stop mode. Then, when Stop mode is requested, FlexCAN performs the following steps:

- Waits to be in Idle state, or else waits for the third bit of Intermission, and then checks it to be recessive.

For Assessment Purpos

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- Sets the LPM_ACK bit in CAN_MCR Register.
- Requests the shutdown of the CHI sub-module clock, while keeping the PE sub-module clock active.

FlexCAN can exit Pretended Networking mode by the following ways:

- The CPU removing the Stop Mode request.
- FlexCAN will wait until Bus Idle or third bit of Intermission state to negate CAN_MCR[LPM_ACK] bit.

The above exit ways can be triggered either by the FlexCAN action upon detecting a wake up event and issuing the respective interrupt, or by the CPU itself upon being waked up by another mean. In consequence, FlexCAN will wait until the Bus Idle state or until the third bit of Intermission state to negate CAN_MCR[LPM_ACK] bit and resume to the Normal mode. This procedure ensures that FlexCAN will be synchronized to the CAN bus after exiting the Pretended Networking mode. The CPU must wait for the CAN_MCR[LPM_ACK] bit to be negated before performing any access to FlexCAN.

When PNET_EN bit in CAN_MCR is asserted, the CPU must disable the Self Wake Up feature by configuring SLF_WAK=0 in CAN_MCR register (see Module Configuration Register).

46.5.12 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, Wake Up Match, Wake Up Timeout, 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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.

The interrupt sources for Pretended Networking (Wake up by Match Flag and Wake up by Timeout Flag) can be read in the CAN_WU_MTC Register and the respective interrupts masks bits are located in CAN_CTRL1_PN Register.

46.5.13 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

- 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.
- If MAXMB in CAN MCR register 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.

46.6 Initialization/application information

This section provide instructions for initializing the FlexCAN module.

46.6.1 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 46-7 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

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
  - If Pretended Networking mode is required, set PNET_EN 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

For Assessment Purposes Only

    - 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)
- If Pretended Networking mode is enabled, configure the necessary registers for selective Wake Up
- Negate the HALT bit in CAN_MCR

After the last step listed above, FlexCAN attempts to synchronize to the CAN bus.

For Assessment Purposes Only

For Assessment Purposes Only

Chapter 47
General-Purpose Input/Output (GPIO)

47.1 Chip-specific General-Purpose Input/Output (GPIO) information

47.1.1 Instantiation Information

Port control and interrupt module features are supported, each 32-pin port will support a single interrupt. Refer to chapter Port Control and Interrupts (PORT) for details of how to control the ports.

47.2 Introduction

The general-purpose input and output (GPIO) module communicates to the processor core via a zero wait state interface 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.

47.2.1 Features

Features of the GPIO module include:

- Port Data Input register visible in all digital pin-multiplexing modes

For Assessment Purposes Only

- Port Data Output register with corresponding set/clear/toggle registers
- Port Data Direction register

NOTE

The GPIO module is clocked by system clock.

47.2.2 Modes of operation

The following table depicts different modes of operation and the behavior of the GPIO module in these modes.

Table 47-1. 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.

47.2.3 GPIO signal descriptions

Table 47-2. 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
PORTD31–PORTD0	General-purpose input/output	I/O
PORTE31–PORTE0	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.

47.2.3.1 Detailed signal description

Table 47-3. GPIO interface-detailed signal descriptions

Signal	I/O	Description
PORTA31–PORTA0	I/O	General-purpose input/output

Table continues on the next page...

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 47-3. GPIO interface-detailed signal descriptions (continued)

Signal	I/O	Description	
PORTB31–PORTB0 PORTC31–PORTC0 PORTD31–PORTD0 PORTE31–PORTE0		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. 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.

47.3 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 each port.

47.3.1 GPIO Register Descriptions

For Assessment Purpo

Freescale Confidential Proprietary - Non-Disclosure Agreement required

47.3.1.1 GPIO Memory Map

Offset	Register	Width (In bits)	Access	Reset value
0h	Port Data Output Register (PDOR)	32	RW	00000000h
4h	Port Set Output Register (PSOR)	32	WORZ	00000000h
8h	Port Clear Output Register (PCOR)	32	WORZ	00000000h
Ch	Port Toggle Output Register (PTOR)	32	WORZ	00000000h
10h	Port Data Input Register (PDIR)	32	RO	00000000h
14h	Port Data Direction Register (PDDR)	32	RW	00000000h
18h	Port Input Disable Register (PIDR)	32	RW	00000000h

47.3.1.2 Port Data Output Register (PDOR)

47.3.1.2.1 Address

Register	Offset
PDOR	0h

47.3.1.2.2 Function

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.

For Assessment Purposes

Freescale Confidential Proprietary - Non-Disclosure Agreement required

47.3.1.2.3 Diagram

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

47.3.1.2.4 Fields

Field	Function
31-0 PDO	Port Data Output Register bits for unbonded pins return a undefined value when read. 00000000000000000000000000000000b - Logic level 0 is driven on pin, provided pin is configured for general-purpose output. 00000000000000000000000000000001b - Logic level 1 is driven on pin, provided pin is configured for general-purpose output.

47.3.1.3 Port Set Output Register (PSOR)

47.3.1.3.1 Address

Register	Offset
PSOR	4h

47.3.1.3.2 Function

This register configures whether to set the fields of the PDOR.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

47.3.1.3.3 Diagram

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

47.3.1.3.4 Fields

Field	Function
31-0 PTSO	Port Set Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 00000000000000000000000000000000b - Corresponding bit in PDORn does not change. 00000000000000000000000000000001b - Corresponding bit in PDORn is set to logic 1.

47.3.1.4 Port Clear Output Register (PCOR)

47.3.1.4.1 Address

Register	Offset
PCOR	8h

47.3.1.4.2 Function

This register configures whether to clear the fields of PDOR.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

47.3.1.4.3 Diagram

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

47.3.1.4.4 Fields

Field	Function
31-0 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: 00000000000000000000000000000000b - Corresponding bit in PDORn does not change. 00000000000000000000000000000001b - Corresponding bit in PDORn is cleared to logic 0.

47.3.1.5 Port Toggle Output Register (PTOR)

47.3.1.5.1 Address

Register	Offset
PTOR	Ch

47.3.1.5.2 Diagram

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

Freescale Confidential Proprietary - Non-Disclosure Agreement required

47.3.1.5.3 Fields

Field	Function
31-0 PTTO	Port Toggle Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 00000000000000000000000000000000b - Corresponding bit in PDORn does not change. 00000000000000000000000000000001b - Corresponding bit in PDORn is set to the inverse of its existing logic state.

47.3.1.6 Port Data Input Register (PDIR)

47.3.1.6.1 Address

Register	Offset
PDIR	10h

47.3.1.6.2 Function

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.

47.3.1.6.3 Diagram

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

Freescale Confidential Proprietary Assessment Purposes Only

47.3.1.6.4 Fields

Field	Function
31-0 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. 00000000000000000000000000000000b - Pin logic level is logic 0, or is not configured for use by digital function. 00000000000000000000000000000001b - Pin logic level is logic 1.

47.3.1.7 Port Data Direction Register (PDDR)

47.3.1.7.1 Address

Register	Offset
PDDR	14h

47.3.1.7.2 Function

The PDDR configures the individual port pins for input or output.

47.3.1.7.3 Diagram

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

47.3.1.7.4 Fields

Field	Function
31-0 PDD	Port Data Direction Configures individual port pins for input or output.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Field	Function
	00000000000000000000000000000000b - Pin is configured as general-purpose input, for the GPIO function. The pin will be high-Z if the port input is disabled in GPIOx_PIDR register. 00000000000000000000000000000001b - Pin is configured as general-purpose output, for the GPIO function.

47.3.1.8 Port Input Disable Register (PIDR)

47.3.1.8.1 Address

Register	Offset
PIDR	18h

47.3.1.8.2 Diagram

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

47.3.1.8.3 Fields

Field	Function
31-0 PID	Port Input Disable 00000000000000000000000000000000b - Pin is configured for General Purpose Input, provided the pin is configured for any digital function. 00000000000000000000000000000001b - Pin is not configured as General Purpose Input. Corresponding Port Data Input Register bit will read zero.

Freescale Confidential Proprietary - Non-Disclosure Agreement required

47.4 Functional description

47.4.1 General-purpose input

The logic state of each pin is available via the Port Data Input registers, provided the corresponding bit in the port input enable register is set, the pin is configured for a digital function and the corresponding Port Control and Interrupt module is enabled.

The input pin synchronizers are shared with the Port Control and Interrupt module, so that if the corresponding Port Control and Interrupt module is disabled, then synchronizers are also disabled. This reduces power consumption when a port is not required for general-purpose input functionality.

47.4.2 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.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

For Assessment Purposes Only

Chapter 48
JTAG Controller (JTAGC)

48.1 Introduction

The JTAGC block provides the means to test chip functionality and connectivity while remaining transparent to system logic when not in test mode. Testing is performed via a boundary scan technique, as defined in the IEEE 1149.1-2001 standard. All data input to and output from the JTAGC block is communicated in serial format.

48.1.1 Block diagram

The following is a simplified block diagram of the JTAG Controller (JTAGC) block. Refer to the chip-specific configuration information as well as Register description for more information about the JTAGC registers.

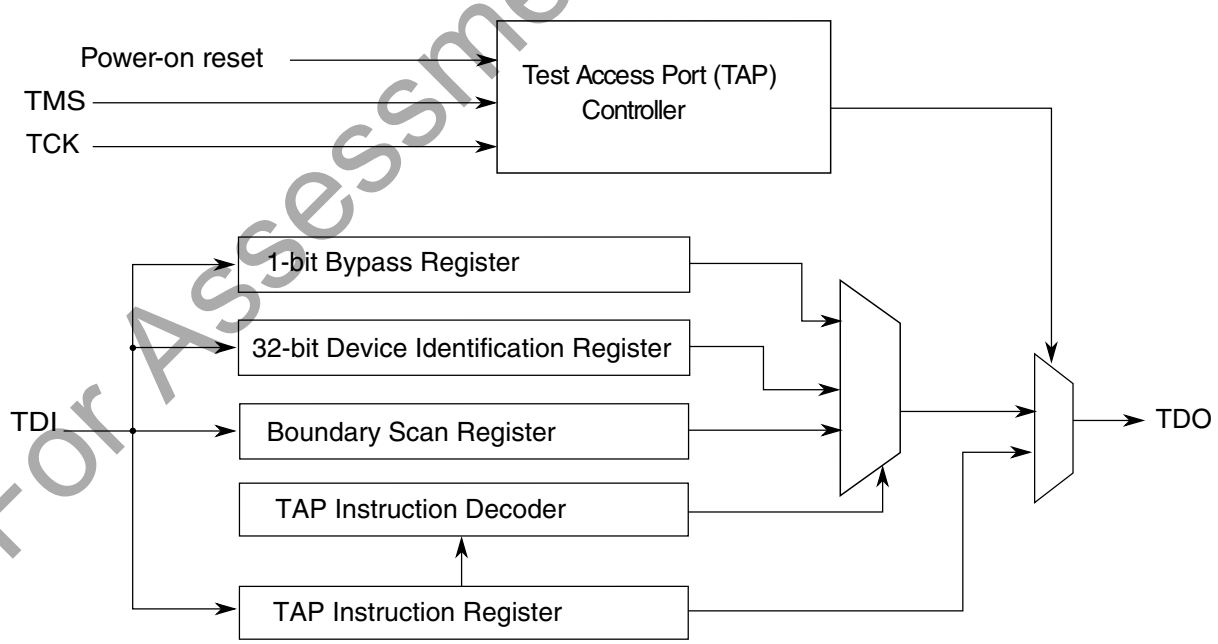


Figure 48-1. JTAG (IEEE 1149.1) block diagram

For Assessment Purposes Only

48.1.2 Features

The JTAGC block is compliant with the IEEE 1149.1-2001 standard, and supports the following features:

- IEEE 1149.1-2001 Test Access Port (TAP) interface
  - 4 pins (TDI, TMS, TCK, and TDO)
- Instruction register that supports several IEEE 1149.1-2001 defined instructions as well as several public and private device-specific instructions. Refer to Table 48-3 for a list of supported instructions.
- Bypass register, boundary scan register, and device identification register.
- TAP controller state machine that controls the operation of the data registers, instruction register and associated circuitry.

48.1.3 Modes of operation

The JTAGC block uses a power-on reset indication as its primary reset signals. Several IEEE 1149.1-2001 defined test modes are supported, as well as a bypass mode.

48.1.3.1 Reset

The JTAGC block is placed in reset when either power-on reset is asserted, or the TMS input is held high for enough consecutive rising edges of TCK to sequence the TAP controller state machine into the Test-Logic-Reset state. Holding TMS high for five consecutive rising edges of TCK guarantees entry into the Test-Logic-Reset state regardless of the current TAP controller state. Asserting power-on reset results in asynchronous entry into the reset state. While in reset, the following actions occur:

- The TAP controller is forced into the Test-Logic-Reset state, thereby disabling the test logic and allowing normal operation of the on-chip system logic to continue unhindered
- The instruction register is loaded with the IDCODE instruction

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

48.1.3.2 IEEE 1149.1-2001 defined test modes

The JTAGC block supports several IEEE 1149.1-2001 defined test modes. A test mode is selected by loading the appropriate instruction into the instruction register while the JTAGC is enabled. Supported test instructions include EXTEST, HIGHZ, CLAMP, SAMPLE and SAMPLE/PRELOAD. Each instruction defines the set of data register(s) that may operate and interact with the on-chip system logic while the instruction is current. Only one test data register path is enabled to shift data between TDI and TDO for each instruction.

The boundary scan register is enabled for serial access between TDI and TDO when the EXTEST, SAMPLE or SAMPLE/PRELOAD instructions are active. The single-bit bypass register shift stage is enabled for serial access between TDI and TDO when the BYPASS, HIGHZ, CLAMP or reserved instructions are active. The functionality of each test mode is explained in more detail in JTAGC block instructions.

48.1.3.3 Bypass mode

When no test operation is required, the BYPASS instruction can be loaded to place the JTAGC block into bypass mode. While in bypass mode, the single-bit bypass shift register is used to provide a minimum-length serial path to shift data between TDI and TDO.

48.2 External signal description

The JTAGC consists of a set of signals that connect to off chip development tools and allow access to test support functions. The JTAGC signals are outlined in the following table and described in the following sections.

Table 48-1. JTAG signal properties

Name	I/O	Function	Reset State	Pull
TCK	Input	Test Clock	—	Down
TDI	Input	Test Data In	—	Up
TDO	Output	Test Data Out	High Z[1]	—
TMS	Input	Test Mode Select	—	Up

1. TDO output buffer enable is negated when the JTAGC is not in the Shift-IR or Shift-DR states. A weak pull may be implemented at the TDO pad for use when JTAGC is inactive.

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

48.2.1 TCK—Test clock input

Test Clock Input (TCK) is an input pin used to synchronize the test logic and control register access through the TAP.

48.2.2 TDI—Test data input

Test Data Input (TDI) is an input pin that receives serial test instructions and data. TDI is sampled on the rising edge of TCK.

48.2.3 TDO—Test data output

Test Data Output (TDO) is an output pin that transmits serial output for test instructions and data. TDO is three-stateable and is actively driven only in the Shift-IR and Shift-DR states of the TAP controller state machine, which is described in TAP controller state machine.

48.2.4 TMS—Test mode select

Test Mode Select (TMS) is an input pin used to sequence the IEEE 1149.1-2001 test control state machine. TMS is sampled on the rising edge of TCK.

48.3 Register description

This section provides a detailed description of the JTAGC block registers accessible through the TAP interface, including data registers and the instruction register. Individual bit-level descriptions and reset states of each register are included. These registers are not memory-mapped and can only be accessed through the TAP.

48.3.1 Instruction register

The JTAGC block uses a 4-bit instruction register as shown in the following figure. The instruction register allows instructions to be loaded into the block to select the test to be performed or the test data register to be accessed or both. Instructions are shifted in through TDI while the TAP controller is in the Shift-IR state, and latched on the falling edge of TCK in the Update-IR state. The latched instruction value can only be changed in the Update-IR and Test-Logic-Reset TAP controller states. Synchronous entry into the

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Test-Logic-Reset state results in the IDCODE instruction being loaded on the falling edge of TCK. Asynchronous entry into the Test-Logic-Reset state results in asynchronous loading of the IDCODE instruction. During the Capture-IR TAP controller state, the instruction shift register is loaded with the value 0b1, making this value the register's read value when the TAP controller is sequenced into the Shift-IR state.

	3	2	1	0
R	0	0	0	1
W	Instruction Code			
Reset:	0	0	0	1

Figure 48-2. Instruction register

48.3.2 Bypass register

The bypass register is a single-bit shift register path selected for serial data transfer between TDI and TDO when the BYPASS, CLAMP, HIGHZ or reserve instructions are active. After entry into the Capture-DR state, the single-bit shift register is set to a logic 0. Therefore, the first bit shifted out after selecting the bypass register is always a logic 0.

48.3.3 Device identification register

The device identification (JTAG ID) register, shown in the following figure, allows the revision number, part number, manufacturer, and design center responsible for the design of the part to be determined through the TAP. The device identification register is selected for serial data transfer between TDI and TDO when the IDCODE instruction is active. Entry into the Capture-DR state while the device identification register is selected loads the IDCODE into the shift register to be shifted out on TDO in the Shift-DR state. No action occurs in the Update-DR state.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Part Revision Number				Design Center						Part Identification Number					
W																
Reset	PRN				DC						PIN					

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Part Identification Number				Manufacturer Identity Code											1
W																
Reset	PIN (contd.)				MIC											1

The following table describes the device identification register functions.

Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 48-2. Device identification register field descriptions

Field	Description
PRN	Part Revision Number. Contains the revision number of the part. Value is 0x0.
DC	Design Center. Indicates the design center. Value is 0x2B.
PIN	Part Identification Number. Contains the part number of the device. Value is 10111110111100100101111111100b.
MIC	Manufacturer Identity Code. Contains the reduced Joint Electron Device Engineering Council (JEDEC) ID. Value is 0x00E.
IDCODE ID	IDCODE Register ID. Identifies this register as the device identification register and not the bypass register. Always set to 1.

48.3.4 Boundary scan register

The boundary scan register is connected between TDI and TDO when the EXTEST, SAMPLE or SAMPLE/PRELOAD instructions are active. It is used to capture input pin data, force fixed values on output pins, and select a logic value and direction for bidirectional pins. Each bit of the boundary scan register represents a separate boundary scan register cell, as described in the IEEE 1149.1-2001 standard and discussed in Boundary scan. The size of the boundary scan register and bit ordering is device-dependent and can be found in the device BSDL file.

48.4 Functional description

This section explains the JTAGC functional description.

48.4.1 JTAGC reset configuration

While in reset, the TAP controller is forced into the Test-Logic-Reset state, thus disabling the test logic and allowing normal operation of the on-chip system logic. In addition, the instruction register is loaded with the IDCODE instruction.

48.4.2 IEEE 1149.1-2001 (JTAG) Test Access Port

The JTAGC block uses the IEEE 1149.1-2001 TAP for accessing registers. This port can be shared with other TAP controllers on the MCU. Ownership of the port is determined by the value of the currently loaded instruction.

For Assessment Purposes C

Freescale Confidential Proprietary - Non-Disclosure Agreement required

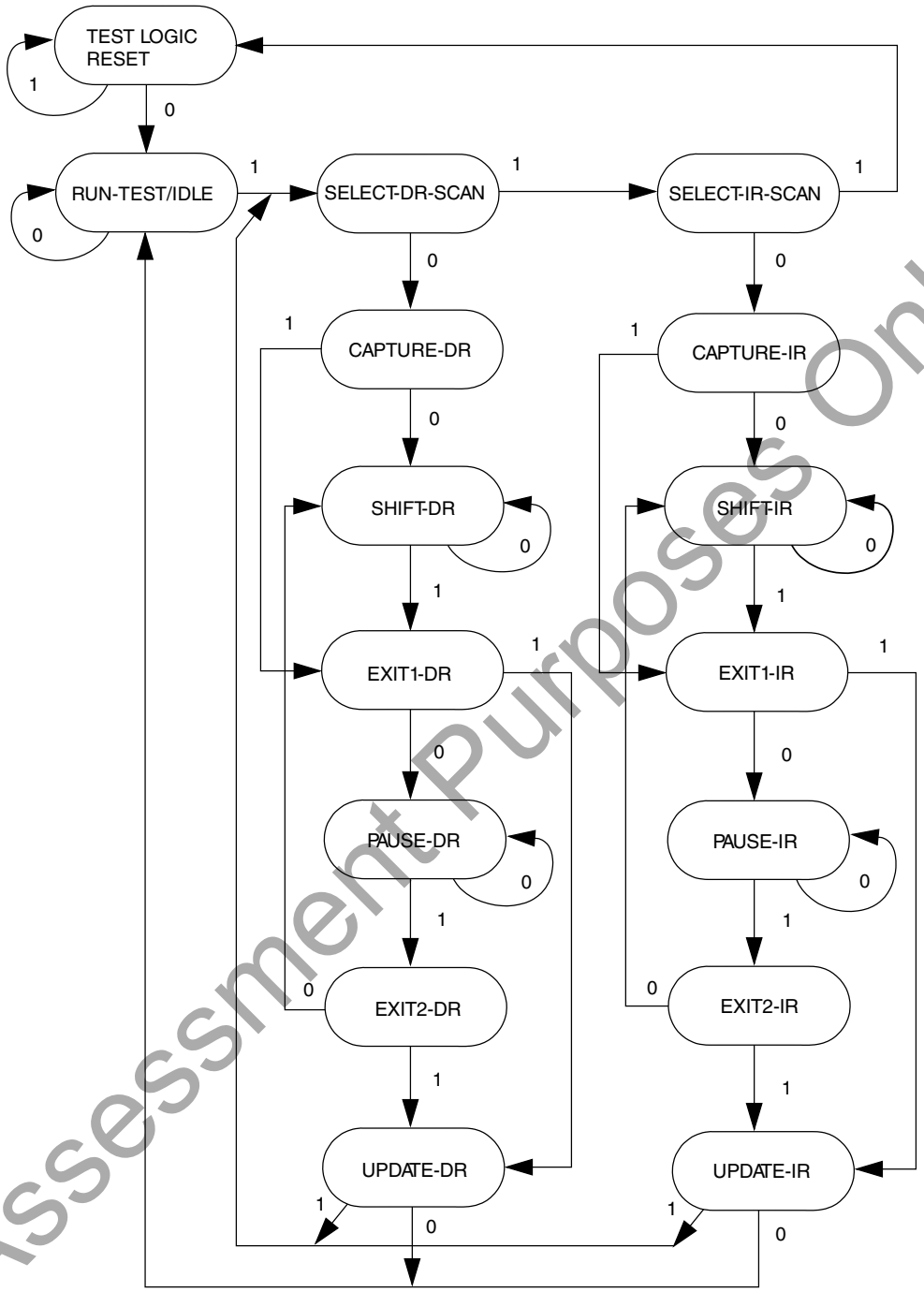
Data is shifted between TDI and TDO though the selected register starting with the least significant bit, as illustrated in the following figure. This applies for the instruction register, test data registers, and the bypass register.

Figure 48-3. Shifting data through a register

48.4.3 TAP controller state machine

The TAP controller is a synchronous state machine that interprets the sequence of logical values on the TMS pin. The following figure shows the machine's states. The value shown next to each state is the value of the TMS signal sampled on the rising edge of the TCK signal. As the following figure shows, holding TMS at logic 1 while clocking TCK through a sufficient number of rising edges also causes the state machine to enter the Test-Logic-Reset state.

For Assessment Purposes Only



The value shown adjacent to each state transition in this figure represents the value of TMS at the time of a rising edge of TCK.

Figure 48-4. IEEE 1149.1-2001 TAP controller finite state machine

For Assessment Purposes Only

48.4.3.1 Enabling the TAP controller

The JTAGC TAP controller is enabled by setting the JTAGC enable to a logic 1 value.

48.4.3.2 Selecting an IEEE 1149.1-2001 register

Access to the JTAGC data registers is achieved by loading the instruction register with any of the JTAGC block instructions while the JTAGC is enabled. Instructions are shifted in via the Select-IR-Scan path and loaded in the Update-IR state. At this point, all data register access is performed via the Select-DR-Scan path.

The Select-DR-Scan path is used to read or write the register data by shifting in the data (LSB first) during the Shift-DR state. When reading a register, the register value is loaded into the IEEE 1149.1-2001 shifter during the Capture-DR state. When writing a register, the value is loaded from the IEEE 1149.1-2001 shifter to the register during the Update-DR state. When reading a register, there is no requirement to shift out the entire register contents. Shifting may be terminated once the required number of bits have been acquired.

48.4.4 JTAGC block instructions

The JTAGC block implements the IEEE 1149.1-2001 defined instructions listed in the following table. This section gives an overview of each instruction; refer to the IEEE 1149.1-2001 standard for more details. All undefined opcodes are reserved.

Table 48-3. 4-bit JTAG instructions

Instruction	Code[3:0]	Instruction summary
IDCODE	0000	Selects device identification register for shift
SAMPLE/PRELOAD	0010	Selects boundary scan register for shifting, sampling, and preloading without disturbing functional operation
SAMPLE	0011	Selects boundary scan register for shifting and sampling without disturbing functional operation
EXTEST	0100	Selects boundary scan register and applies preloaded values to output pins. **NOTE:** Execution of this instruction asserts functional reset.
Factory debug reserved	0101	Intended for factory debug only
Factory debug reserved	0110	Intended for factory debug only
Factory debug reserved	0111	Intended for factory debug only
ARM JTAG-DP Reserved	1000	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.
HIGHZ	1001	Selects bypass register and three-states all output pins. **NOTE:** Execution of this instruction asserts functional reset.
ARM JTAG-DP Reserved	1010	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.

Table continues on the next page...

nt Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

Table 48-3. 4-bit JTAG instructions (continued)

Instruction	Code[3:0]	Instruction summary
ARM JTAG-DP Reserved	1011	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.
CLAMP	1100	Selects bypass register and applies preloaded values to output pins. **NOTE:** Execution of this instruction asserts functional reset.
ARM JTAG-DP Reserved	1110	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.
BYPASS	1111	Selects bypass register for data operations

48.4.4.1 IDCODE instruction

IDCODE selects the 32-bit device identification register as the shift path between TDI and TDO. This instruction allows interrogation of the MCU to determine its version number and other part identification data. IDCODE is the instruction placed into the instruction register when the JTAGC block is reset.

48.4.4.2 SAMPLE/PRELOAD instruction

The SAMPLE/PRELOAD instruction has two functions:

- The SAMPLE portion of the instruction obtains a sample of the system data and control signals present at the MCU input pins and just before the boundary scan register cells at the output pins. This sampling occurs on the rising edge of TCK in the Capture-DR state when the SAMPLE/PRELOAD instruction is active. The sampled data is viewed by shifting it through the boundary scan register to the TDO output during the Shift-DR state. Both the data capture and the shift operation are transparent to system operation.
- The PRELOAD portion of the instruction initializes the boundary scan register cells before selecting the EXTEST or CLAMP instructions to perform boundary scan tests. This is achieved by shifting in initialization data to the boundary scan register during the Shift-DR state. The initialization data is transferred to the parallel outputs of the boundary scan register cells on the falling edge of TCK in the Update-DR state. The data is applied to the external output pins by the EXTEST or CLAMP instruction. System operation is not affected.

For Assessment Purposes Only

Freescale Confidential Proprietary - Non-Disclosure Agreement required

48.4.4.3 SAMPLE instruction

The SAMPLE instruction obtains a sample of the system data and control signals present at the MCU input pins and just before the boundary scan register cells at the output pins. This sampling occurs on the rising edge of TCK in the Capture-DR state when the SAMPLE instruction is active. The sampled data is viewed by shifting it through the boundary scan register to the TDO output during the Shift-DR state. There is no defined action in the Update-DR state. Both the data capture and the shift operation are transparent to system operation.

48.4.4.4 EXTEST External test instruction

EXTEST selects the boundary scan register as the shift path between TDI and TDO. It allows testing of off-chip circuitry and board-level interconnections by driving preloaded data contained in the boundary scan register onto the system output pins. Typically, the preloaded data is loaded into the boundary scan register using the SAMPLE/PRELOAD instruction before the selection of EXTEST. EXTEST asserts the internal system reset for the MCU to force a predictable internal state while performing external boundary scan operations.

48.4.4.5 HIGHZ instruction

HIGHZ selects the bypass register as the shift path between TDI and TDO. While HIGHZ is active all output drivers are placed in an inactive drive state (e.g., high impedance). HIGHZ also asserts the internal system reset for the MCU to force a predictable internal state.

48.4.4.6 CLAMP instruction

CLAMP allows the state of signals driven from MCU pins to be determined from the boundary scan register while the bypass register is selected as the serial path between TDI and TDO. CLAMP enhances test efficiency by reducing the overall shift path to a single bit (the bypass register) while conducting an EXTEST type of instruction through the boundary scan register. CLAMP also asserts the internal system reset for the MCU to force a predictable internal state.

For Assessment Purposes Only

48.4.4.7 BYPASS instruction

BYPASS selects the bypass register, creating a single-bit shift register path between TDI and TDO. BYPASS enhances test efficiency by reducing the overall shift path when no test operation of the MCU is required. This allows more rapid movement of test data to and from other components on a board that are required to perform test functions. While the BYPASS instruction is active the system logic operates normally.

48.4.5 Boundary scan

The boundary scan technique allows signals at component boundaries to be controlled and observed through the shift-register stage associated with each pad. Each stage is part of a larger boundary scan register cell, and cells for each pad are interconnected serially to form a shift-register chain around the border of the design. The boundary scan register consists of this shift-register chain, and is connected between TDI and TDO when the EXTEST, SAMPLE, or SAMPLE/PRELOAD instructions are loaded. The shift-register chain contains a serial input and serial output, as well as clock and control signals.

48.5 Initialization/Application information

The test logic is a static logic design, and TCK can be stopped in either a high or low state without loss of data. However, the system clock is not synchronized to TCK internally. Any mixed operation using both the test logic and the system functional logic requires external synchronization.

To initialize the JTAGC block and enable access to registers, the following sequence is required:

1. Place the JTAGC in reset through TAP controller state machine transitions controlled by TMS
2. Load the appropriate instruction for the test or action to be performed

For Assessment Purposes Only

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, ColdFire+, C-Ware, Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorivva, StarCore, Symphony, and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, Layerscape, MagniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SafeAssure, SafeAssure logo, SMARTMOS, Tower, TurboLink, Vybrid, and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2014–2015 Freescale Semiconductor, Inc.

For Assessment Purposes Only

Document Number S32K144RM
Revision 1 Draft H, 02/2016