MKW01Z128

Sub 1 GHz Low Power Transceiver plus Microcontroller
Reference Manual

Document Number: MKW01xxRM
Rev. 2
3/2014

freescale™ semiconductor

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

Chapter 1
MKW01Z128 Introduction and Chip Configuration

Chapter 2
MKW01Z128 Pins and Connections

Chapter 3 Signal Multiplexing and Signal Descriptions

Chapter 4
System Considerations

Chapter 5
Sub 1 GHz Transceiver Architecture Description

Chapter 6
Transceiver Operating Modes

Chapter 7
Transceiver Digital Control and Communications

Contents

About This Book

This manual details the MKW01, which is a highly-integrated, cost-effective, system-in-package (SIP), sub-1 GHz wireless node solution with an FSK, GFSK, MSK, or OOK modulation-capable transceiver and low-power Kinetis microcontroller. The highly integrated RF transceiver operates over a wide frequency range including 315 MHz, 433 MHz, 470 MHz, 868 MHz, 915 MHz, 928 MHz, and 955 MHz in the license-free Industrial, Scientific and Medical (ISM) frequency bands.

Audience

This manual is intended for system designers.

Revision History

The following table summarizes revisions to this document since the previous release (Rev 1.0).

Revision History

Location	Revision
Throughout	Replaced pinout. Minor typographic changes and clarifications throughout.

Definitions, Acronyms, and Abbreviations

The following list defines the acronyms and abbreviations used in this document.

ACK	Acknowledgement Frame
API	Application Programming Interface
BB	Baseband
CCA	Clear Channel Assessment
CRC	Cyclical Redundancy Check
DCD	Differential Chip Decoding
DME	Device Management Entity
FCS	Frame Check Sequence
FFD	Full Function Device
FFD-C	Full Function Device Coordinator
FLI	Frame Length Indicator
GTS	Guaranteed Time Slot
HW	Hardware
IRQ	Interrupt Request
ISR	Interrupt Service Routine
LO	Local Oscillator
MAC	Medium Access Control
MCPS	MAC Common Part Sublayer
MCU	Microcontroller Unit
MLME	MAC Sublayer Management Entity
MSDU	MAC Service Data Unit
NWK	Network
PA	Power Amplifier
PAN	Personal Area Network
PANID	PAN Identification
PHY	PHYsical Layer
PIB	PAN Information Base
PPDU	PHY Protocol Data Unit
PSDU	PHY Service Data Unit
RF	Radio Frequency
RFD	Reduced Function Device
SAP	Service Access Point
SFD	Start of Frame Delimiter

SPI	Serial Peripheral Interface
SSCS	Service Specific Convergence Layer
SW	Software
VCO	Voltage Controlled Oscillator

References

The following sources were referenced to produce this book:

[1] IEEE 802.15.4 Standard
[2] Freescale MKW01xx Data Sheet

Chapter 1
MKW01Z128 Introduction and Chip Configuration

Kinetis is the most scalable portfolio of low power, mixed-signal ARM®Cortex™ MCUs in the industry. Kinetis MCU families are peripheral- and software-compatible devices. Each family offers excellent performance, memory and feature scalability with common peripherals, memory maps, and packages providing easy migration both within and between families.

Kinetis MCUs are built from Freescale’s innovative 90 nm thin film storage (TFS) flash technology with unique FlexMemory. Kinetis MCU families combine the latest low-power innovations and high performance, high precision mixed-signal capability with a broad range of connectivity, human-machine interface, and safety & security peripherals.Kinetis MCUs are supported by a market-leading enablement bundle from Freescale and numerous ARM 3rd party ecosystem partners.

Kinetis W-series devices all contain wireless connectivity options spanning across frequency bands and standards.

Table 1-1. Kinetis W-Series devices

Family	Frequency Band
KW0x	Sub-Gigahertz
KW2x	2.4 GHz
KW3x	Reserved

KW01 devices also have these features:

- Core:
  - ARM Cortex-M0+ Cores delivering single-cycle access memories, 48 MHz CPU frequency
  - Up to 16-channel DMA for peripheral and memory servicing with minimal CPU intervention
  - Broad range of performance levels rated at maximum CPU frequencies starting at 48 MHz
- Ultra-low power:
  - Multiple low power operating modes for optimizing peripheral activity and wakeup times for extended battery life.
  - Low–leakage wakeup unit, low power timer, and low power RTC for additional low power flexibility
  - Industry-leading fast wakeup times
- Memory: 16 KB RAM, 128 KB flash
- Mixed-signal analog:

  - Fast, high precision 16-bit ADCs, 12-bit DACs, high speed comparators and an internal voltage reference. Powerful signal conditioning, conversion and analysis capability with reduced system cost
- Human Machine Interface (HMI):
  - Capacitive Touch Sensing Interface with full low-power support and minimal current adder when enabled
- Connectivity and Communications:
  - UARTs with ISO7816, CEA709.1-B (LON), and IrDA support, I2C, and DSPI
- Reliability, Safety and Security:
  - Hardware cyclic redundancy check engine for validating memory contents/ communication data and increased system reliability
  - Independent-clocked computer operating properly (COP) for protection against code runaway in fail-safe applications
  - External watchdog monitor
- Timing and Control:
  - Programmable Interrupt Timer for RTOS task scheduler time base or trigger source for ADC conversion and programmable delay block
- System:
  - Wide operating voltage range from 1.8 V to 3.6 V with flash programmable down to 1.8 V with fully functional flash and analog peripherals
  - Ambient operating temperature ranges from –40°C to 85°C

1.1 KW01 family introduction

The KW01 family is the entry point into the Kinetis W-Series portfolio. The K01W is a single-chip solution combining an ARM Cortex-M0+ microcontroller and a sub-GHz ISM band radio front-end device.

Devices contain 128 KB of flash and 16 KB of SRAM in an 8 x 8 mm 56 LGA package. Standard features include a rich suite of analog, communication, timing and control peripherals. Additionally, flexible low-power capabilities and innovative FlexMemory help to solve many of the major pain points for system implementation.

1.2 Ordering information

Table 1-2 lists the available devices in the MKW01 family.

Table 1-2. Devices in the MKW01 Family

Device	Operating Temp Range (TA.)	Package	Memory Options	Description
MKW01Z128CHN	–40° to 85° C	LGA	16 KB RAM, 128 KB flash	The primary target market is communications for last mile metering, sub metering and associated devices such as concentrators. The feature set will also allow it to serve for wireless sensor networks in building control and automation.

1.3 General platform features

- ARM Cortex-M0+ Core
- Sub-1 GHz in-package transceiver
- Multiple power saving modes
- 1.8 V to 3.6 V operating voltage with on-chip voltage regulators
- –40°C to +85°C temperature range
- Low external component count
- Supports single crystal (32 MHz typical) clock source operation or dual crystal operation
- Versatile software solutions
- 56-pin LGA (8x8 mm) Package

1.4 MCU features

- Core:
  - ARM Cortex-M0+ 1.77 CoreMark/MHz from single-cycle access memories, 48 MHz CPU frequency
  - 4-channel DMA for peripheral and memory servicing with minimal CPU intervention
  - CPU frequencies up to 48 MHz
- Ultra-low power:
  - Multiple low power operating modes for optimizing peripheral activity and wakeup times for extended battery life.
  - Low–leakage wakeup unit and low power timer for time keeping function
  - Industry-leading fast wakeup times
- Memory:
  - Up to 128 KB total flash memory (128KB)
- Mixed-signal analog:
  - Fast, high precision 16-bit ADCs, and internal high speed comparators. Powerful signal conditioning, conversion and analysis capability with reduced system cost
- Human Machine Interface (HMI):
  - Capacitive Touch Sensing Interface with full low-power support and minimal current adder when enabled

- Connectivity and Communications:
  - Three UARTs, two SPIs, and two I^2C
- Reliability, Safety and Security:
  - Hardware cyclic redundancy check engine for validating memory contents/ communication data and increased system reliability
  - Independent-clocked computer operating properly (COP) for protection against code runaway in fail-safe applications
- Timing and Control:
  - Powerful timer modules that support general-purpose, PWM, and motor control functions
  - Programmable Interrupt Timer for RTOS task scheduler time base or trigger source for ADC conversion and programmable delay block
- System:
  - Wide operating voltage range from 1.8 V to 3.6 V with flash programmable down to 1.8 V with fully functional flash and analog peripherals
  - Ambient operating temperature ranges from –40°C to 85°C

1.5 RF transceiver features

- High Sensitivity: down to –120 dBm at 1.2 kbps
- High Selectivity: 16-tap FIR Channel Filter
- Bullet-proof front end: IIP3 = –18 dBm, IIP2 = +35 dBm, 80 dB Blocking Immunity, no Image Frequency response
- Low current: RX = 16 mA, 100 nA register retention
- Programmable Pout : –18 to +17 dBm in 1 dB steps
- Constant RF performance over voltage range of chip
- FSK bit rates up to 600 kbps
- Fully integrated synthesizer with a resolution of 61 Hz
- FSK, GFSK, MSK, GMSK and OOK modulations
- Built-in Bit Synchronizer performing Clock recovery
- Incoming Sync Word Recognition
- Automatic RF Sense with ultra-fast AFC
- Packet engine with CRC, AES-128 encryption and 66-byte FIFO
- Built-in temperature sensor and Low battery indicator
- 32 MHz (typical) crystal oscillator clock source

1.6 Software solutions

Freescale will support the MKW01Z128 platform with several software solutions:

- A radio utility GUI will be available that allows testing of various features and setting registers. A firmware-based connectivity test will allow a limited set of testing controlled with a terminal emulator on any computer.
- SMAC (Simple Media Access Controller) — This codebase provides simple communication and test apps based on drivers/PHY utilities available as source code. This environment is useful for hardware and RF debug, hardware standards certification, and developing proprietary applications.
- MAC/PHY (Media Access Control/Physical) for IEEE 802.15.4g/e — This release was developed primarily for the ZigBee Alliance specified Home Energy Management Systems for the Japanese application space.
- Additional software will be available through 3rd party providers.

1.7 System overview

Figure 1-1 shows a simplified block diagram of the MKW01.

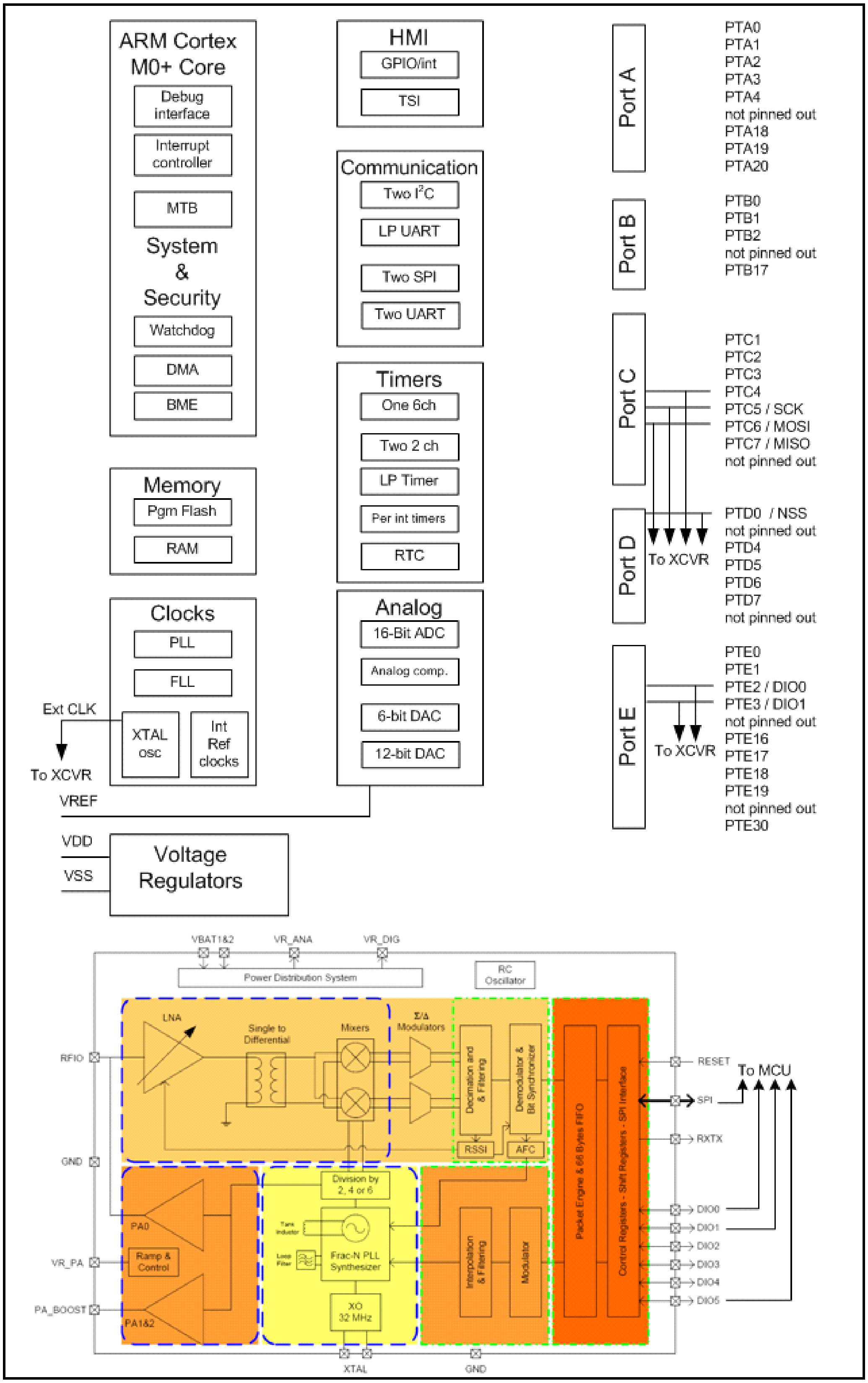


Figure 1-1. MKW01 system level block diagram

1.7.1 Transceiver overview

The transceiver (see Figure 1-1) is a single-chip integrated circuit ideally suited for today's high performance ISM band RF applications. Its advanced features set, including state of the art packet engine, greatly simplifies system design while the high level of integration reduces the external RF component bill of material (BOM) to a handful of passive de-coupling and matching components. It is intended for use as a high-performance, low-cost FSK and OOK RF transceiver for robust, frequency agile, half-duplex bidirectional RF links.

The MKW01 is intended for applications over a wide frequency range, including the 433 MHz and 868 MHz European and the 902–928 MHz North American and Japan ISM bands. Coupled with a link budget in excess of 135 dB, the transceiver advanced system features include a 66 byte TX/RX FIFO, configurable automatic packet handler, listen mode, temperature sensor and configurable DIOs which greatly enhance system flexibility while at the same time significantly reducing MCU requirements. The transceiver complies with both ETSI and FCC regulatory requirements.

The major RF communication parameters of the MKW01 transceiver are programmable and most can be dynamically set. This feature offers the unique advantage of programmable narrow-band and wide-band communication modes without the need to modify external components. The transceiver is also optimized for low power consumption while offering high RF output power and channelized operation.

1.7.2 MCU overview

The in-package Kinetis L series 48 MHz MCU features an ARM Cortex M0+, 16 KB Ram and 128 KB flash. The RF transceiver is controlled through the MCU SPI port which is dedicated to the RF device interface. Two of the transceiver status IO lines are also directly connected to the MCU GPIO to monitor the transceiver operation. In addition, the transceiver reset and additional status can be connected to the MCU through external connections.

1.7.2.1 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 1-3. Module functional categories

Module category	Description
ARM Cortex-M0+ core	
System	• System integration module • Power management and mode controllers — Multiple power modes available based on run, wait, stop, and powerdown modes • Low-leakage wakeup unit • Miscellaneous control module • Crossbar switch • Peripheral bridge • Direct memory access (DMA) controller with multiplexer to increase available DMA requests • External watchdog monitor • Watchdog

Table 1-3. Module functional categories (continued)

Module category	Description
Memories	Internal memories include: • Up to 128KB program flash memory • Up to 16KB SRAM
Clocks	• Multiple clock generation options available from internally- and externally-generated clocks • System oscillator from transceiver to provide clock source for the MCU • 32 kHz RTC oscillator
Security	• Cyclic Redundancy Check module for error detection
Analog	• 16-bit analog-to-digital converter • Internal Comparator with internal 6-bit DAC for reference • 12-bit DAC with DMA support and two 16-bit buffers
Timers	• Low Power Timer/PWM (TPM) modules • One 6-channel TPM • Two 2-channel TPMs • 2-channel periodic interrupt timer • Real-time clock • Low-power timer • System tick timer
Communications	• 2x internal serial peripheral interface • 2x inter-integrated circuit (I^2C) • 3x UART
Human-Machine Interfaces (HMI)	• General purpose input/output controller • Capacitive touch sense input interface enabled in hardware

1.7.2.2 ARM Cortex-M0 core modules

The following core modules are available on this device.

Table 1-4. Core modules

Module	Description
ARM Cortex-M0+	The ARM Cortex-M0+ is the newest member of the Cortex M Series of processors targeting microcontroller applications focused on very cost sensitive, deterministic, interrupt driven environments. The Cortex M0+ processor is based on the ARMv6 Architecture and Thumb®-2 ISA and is 100% instruction set compatible with its predecessor, the Cortex-M0 core, and upward compatible to Cortex-M3 and M4 cores.
NVIC	The ARMv6-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.

Table 1-4. Core modules (continued)

Module	Description
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.
Single-cycle I/O Port	For high-speed, single-cycle access to peripherals, the Cortex-M0+ processor implements a dedicated single-cycle I/O port.
Debug interfaces	Most of this device's debug is based on the ARM CoreSight™ architecture. One debug interface is supported: • Serial Wire Debug (SWD)

1.7.2.3 System modules

The following system modules are available on this device.

Table 1-5. System modules

Module	Description
System integration module (SIM)	The SIM includes integration logic and several module configuration settings.
System mode controller	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.
Power management controller (PMC)	The PMC provides the user with multiple power options. Multiple modes are supported 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.
Low-leakage wakeup unit (LLWU)	The LLWU module allows the device to wake from low leakage power modes (LLS and VLLS) through various internal peripheral and external pin sources.
Peripheral bridge	The peripheral bridge 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 4 for the DMA controller.
Direct memory access (DMA) controller	The DMA controller provides programmable channels with transfer control descriptors for data movement via dual-address transfers for 8-, 16- and 32-bit data values.
Computer operating properly watchdog (WDOG)	The WDOG monitors internal system operation and forces a reset in case of failure. It can run from an independent 1 kHz low power oscillator with a programmable refresh window to detect deviations in program flow or system frequency.

1.7.2.4 Memories and memory interfaces

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

Table 1-6. Memories and memory interfaces

Module	Description
Flash memory	Program flash memory — up to 128 KB of the non-volatile flash memory that can execute program code
Flash memory controller	Manages the interface between the device and the on-chip flash memory.
SRAM	Up to 16 KB internal system RAM.

1.7.2.5 Clock modules

The following clock modules are available on this device.

Table 1-7. Clock modules

Module	Description
Multi-clock generator (MCG)	The MCG, controlled by an internal or external (such as the CLKOUT from the transceiver) reference oscillator, provides several clock sources for the MCU that include: • Phase-locked loop (PLL). Voltage-controlled oscillator (VCO) • Frequency-locked loop (FLL). Digitally-controlled oscillator (DCO) • Internal reference clocks. Can be used as a clock source for other on-chip peripherals
System oscillator	The system oscillator, in conjunction with an external crystal or resonator, generates a reference clock for the MCU.

1.7.2.6 Security and integrity module

The following security and integrity module is available on this device.

Table 1-8. Security and integrity module

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.

1.7.2.7 Analog modules

The following analog modules are available on this device.

Table 1-9. Analog Modules

Module	Description
16-bit analog-to-digital converters (ADC)	16-bit successive-approximation ADC
Internal analog comparators	Compares two analog input voltages, one of which can be a reference provided by the internal 6-bit DAC, across the full range of the supply voltage.
6-bit digital-to-analog converters (DAC)	64-tap resistor ladder network which provides a selectable voltage reference for analog comparator.

1.7.2.8 Timer modules

The following timer modules are available on this device.

Table 1-10. Timer modules

Module	Description
Timer/PWM module (TPM)	Selectable TPM clock mode • Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128 • 16-bit free-running counter or modulo counter with counting be up or updown • Six configurable channels for input capture, output compare, or edge-aligned PWM mode • Support the generation of an interrupt and/or DMA request per channel • Support the generation of an interrupt and/or DMA request when the counter overflows • Support selectable trigger input to optionally reset or cause the counter to start incrementing. • Support the generation of hardware triggers when the counter overflows and per channel
Periodic interrupt timers (PIT)	• Four general purpose interrupt timers • Interrupt timers for triggering ADC conversions • 32-bit counter resolution • Clocked by system clock frequency • DMA support
Low-power timer (LPTimer)	• Selectable clock for prescaler/glitch filter of 1 kHz (internal LPO), 32.768 kHz (external crystal), 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

1.7.2.9 Radio

Table 1-11. Radio transceiver

Module	Description
Sub-GHz transceiver	• A highly integrated ISM band transceiver for FSK and OOK packet or continuous data.

1.7.2.10 Communication interfaces

The following wired communication interfaces are available on this device.

Table 1-12. Communication interfaces

Module	Description
Internal serial peripheral interface (SPI)	Synchronous serial bus for communication to an external device
Inter-integrated circuit (I2C)	Allows communication between a number of devices. Also supports the System Management Bus (SMBus) Specification, version 2.
Universal asynchronous receiver/transmitters (UART)	Asynchronous serial bus communication interface with programmable 8- or 9-bit data format

1.7.2.11 Human-machine interfaces

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

Table 1-13. 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. All GPIO pins have 5 V tolerance.
Capacitive touch sense input (TSI)	Contains up to 10 channel inputs for capacitive touch sensing applications. Operation is available in low-power modes via interrupts.

1.7.2.12 System Device Identification Register

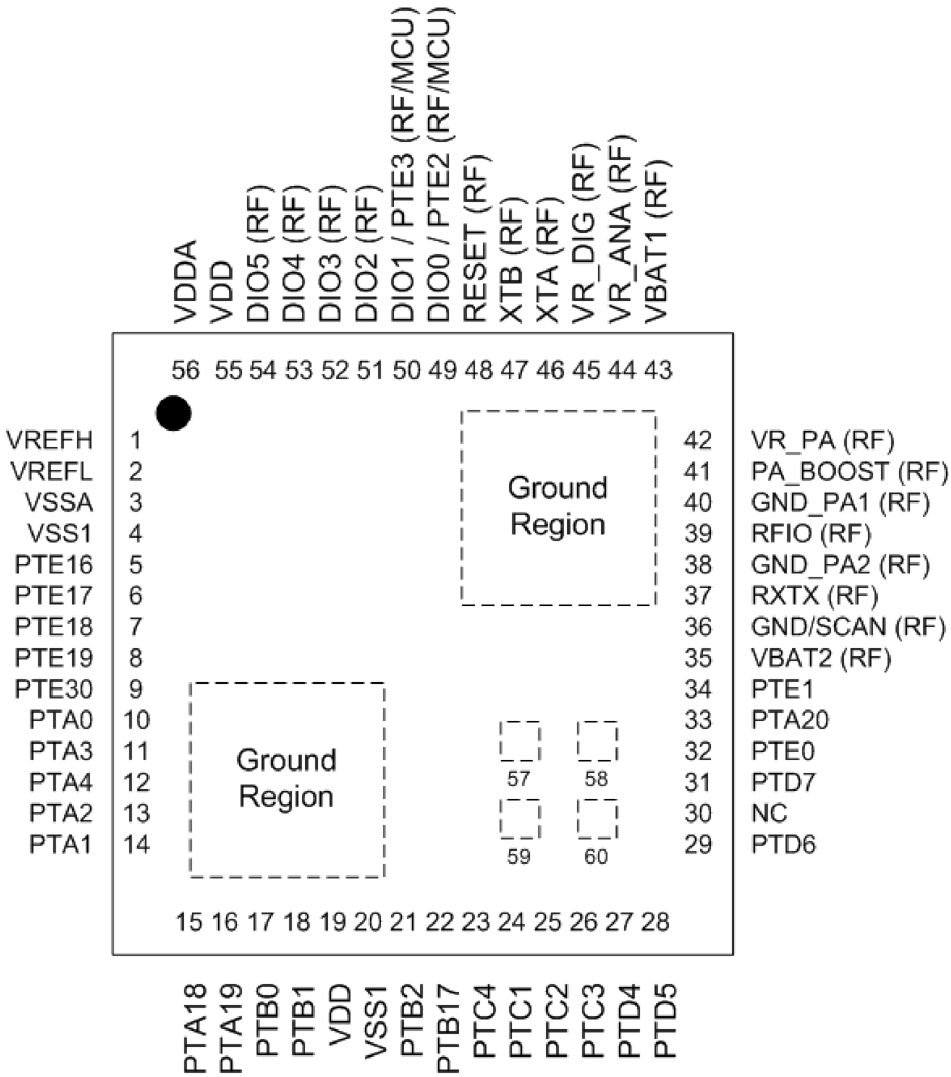
The system device identification register described in chapter 8.2.6 contains device specific information factory programmed into the in-package MCU die.

Table 1-14. Device-Specific Values

Field ID	Value
FAMID	0001
SUBFAMID	0111
SERIESID	0001
SRAMSIZE	0101
REVID	0001
DIEID	01010
PINID	0010

Chapter 2
MKW01Z128 Pins and Connections

2.1 Device pin assignment



2 ground regions and 4 pins on bottom:

57	MISO / PTC7 (RF/MCU)
58	NSS / PTD0 (RF/MCU)
59	SCK / PTC5 (RF/MCU)
60	MOSI / PTC6 (RF/MCU)

Figure 2-1. MKW01Z128 pinout

2.2 Pin definitions

Table 2-1 details the MKW01Z128 pinout and functionality.

Table 2-1. Pin Function Description (Sheet 1 of 5)

Pin #	Pin Name	Type	Description	Functionality
1	VREFH	Input	MCU high reference voltage for ADC	
2	VREFL	Input	MCU low reference voltage for ADC	
3	VSSA	Power Input	MCU ADC Ground	Connect to ground
4	VSS	Power Input	MCU Ground	Connect to ground
5	PTE16/ADC0_DP1/ADCO_SE1/SPI0_PCS0/TPM/UART2_TX	Digital Input / Output	MCU Port E Bit 16 / ADC0 Single Ended analog channel input DP1/ ADC0 Single Ended analog channel input SE1 / SPI module 0 PCS0 / TPM module Clock In 0 / UART2_TX	
6	PTE17/ADC0_DM1/ADCO_SE5a/SPI0_SCK/ TPM_CLKIN1/UART2_RX/LPTMR0_ALT3	Digital Input / Output	MCU Port E Bit 17 / ADC0 Single Ended analog channel input DM1/ ADC0 Single Ended analog channel input 5a / SPI module 0 SCK / TPM module Clock In 1 / UART2_RX / Low Power Timer Module 0 ALT3	
7	PTE18/ADC0_DP2/ADC0_SE2/SPI0_MOSI/I2C0_SDA/SPI0_MISO	Digital Input / Output	MCU Port E Bit 18 / ADC0 Single Ended analog channel input DP2/ ADC0 Single Ended analog channel input 2 / SPI module 0 MOSI / I2C0 Bus Data / SPI module 0 MISO	
8	PTE19/ADC0_DM2/ ADC0_SE6a/SPI0_MISO /I2C0_SCL/SPI0_MOSI	Digital Input / Output	MCU Port E Bit 19 / ADC0 Single Ended analog channel input DM2/ ADC0 Single Ended analog channel input 6a / SPI module 0 MISO / I2C0 Bus Clock / SPI module 0 MOSI	
9	PTE30/DAC0_OUT/ADCO_SE23/ CMP0_IN4/TPM0_CH3/TPM_CLKIN1	Digit-l Input / Output	MCU Port E Bit 30 / DAC0 Output/ ADC0 Single Ended analog channel input 23 / Comparator 0 Analog Voltage Input 4/ TPM Timer module 0 Channel 3 / TPM module Clock In 1	
10	PTA0/SWD_CLK/TSI0_CH1/TPM0_CH5	Digital Input / Output	MCU Port A Bit 0 / Serial Wire Data Clock / Touch Screen Interface Channel 1/ TPM module 0 Channel 5	
11	PTA3/SWD_DIO/TSI0_CH4/I2C1_SCL/TPM0_CH0	Digital Input / Output	MCU Port A Bit 3 / Serial Wire Data DIO / Touch Screen Interface Channel 4 / I2C1 Bus Clock / TPM module 0 Channel 0	
12	PTA4/NMI_b/TSI0_CH5/I2C1_SDA/TPM0_CH1	Digital Input / Output	MCU Port A Bit 4/ / Non Maskable Interrupt_ b/Touch Screen Interface Channel 5 /I2C1 Bus Data / TPM module 0 Channel 1	

Table 2-1. Pin Function Description (Sheet 2 of 5)

Pin #	Pin Name	Type	Description	Functionality
13	PTA2/TSI0_CH3/UART0_TX/ TPM2_CH1	Digital Input / Output	MCU Port A Bit 2/Touch Screen Interface Channel 3/UART module 0 Transmit / TPM module 2 Channel 1	
14	PTA1/TSI0_CH2/UART0_RX/ TPM2_CH0	Digital Input / Output	MCU Port A Bit 1/Touch Screen Interface Channel 2/UART module 0 Receive / TPM module Channel 0	
15	PTA18/EXTAL0/UART1_RX/ TPM_CLKIN0	Digital Input / Output	MCU Port A Bit 18 / EXTAL0/ UART module 1 Receive / TPM module Clock In 0	
16	PTA19/XTAL0/UART1_TX/ TPM_CLKIN1/LPTMR0_ALT1	Digital Input / Output	MCU Port A Bit 19 / XTAL0/ UART module 1 Transmit / TPM module Clock In 1 /Low Power Timer module 0 ALT1	
17	PTB0/ADC0_SE8/TSI0_CH0/ LLWU_P5/I2C0_SCL/ TPM1_ CH0	Digital Input / Output	MCU Port B Bit 0 / ADC0 Single Ended analog channel input SE8 / Touch Screen Interface Channel 0/ Low Leakage Wake Up Port 5 / I2C0 Bus Clock / TPM module 1 Channel 0	
18	PTB1/ADCO_SE9/TSI0_CH6/ I2C0_SDA/ TPM1_CH1	Digital Input / Output	MCU Port B Bit 1 / ADC0 Single Ended analog channel input SE9 / Touch Screen Interface Channel 6 / I2C0 Bus Data / TPM module 1 Channel 1	
19	VDD	Power Input	MCU VDD supply input	Connect to system VDD supply
20	VSS	Power Input	MCU Ground	Connect to ground
21	PTB2/ADC0_SE12/TSI0_ CH7/I2C0_SCL/TPM2_CH0	Digital Input/ Output	MCU Port B Bit 2 / ADC0 Single Ended analog channel input SE12 / Touch Screen Interface Channel 7 / I2C0 Bus Clock / TPM Timer module 2 Channel 0	
22	PTB17/TSI0_CH10/SPI1_ MISO/UART0_TX/TPM_ CLKIN1/SPI1_MOSI	Digital Input/ Output	MCU Port B Bit 17 / Touch Screen Interface Channel 10/SPI1 MOSI or MISO/UART0 TX / TPM timer clock	
23	PTC4/LLWU_P8/SPI0_PCS0/ UART1_TX/TPM0_CH3	Digital Input / Output	MCU Port C bit 4 / Low leakage Wake Up port 8 / SPI0 Chip Select / UART1 TX / TPM Timer module 0 channel 3	
24	PTC1/ADC0_SE15/TSI0_ CH14/LLWU_P6/RTC_CLKIN/ I2C1_SCL/TPM0_CH0	Digital Input Output / Analog Input	MCU Port C Bit 1 /ADC0 Single Ended analog channel input SE15/ Touch Screen Interface Channel 14/ Low Leakage Wake Up Port 6 / Real Time Counter Clock Input/ IC1 Bus Clock/ TPM module 0 Channel 0	
25	PTC2/ADC0_SE11/TSI0_ CH15/I2C1_SDA/TPM0_CH1	Digital Input / Output / Analog Input	MCU Port C Bit 2 / ADC0 Single Ended analog channel input SE11/ / Touch Screen Interface Channel 15 / I2C1 Bus Data / TPM module 0 Channel 1	

Table 2-1. Pin Function Description (Sheet 3 of 5)

Pin #	Pin Name	Type	Description	Functionality
26	PTC3/LLWU_P7/UART1_RX/TPM0_CH2/CLKOUTa	Digital Input / Output	MCU Port C Bit 3 / Low Leakage Wake Up Port 7 / UART module 1 Receive / TPM module 0 Channel 2/ Clock OutA	
27	PTD4/LLWU_P14/SPI1_PCS0/UART2_RX/TPM0_CH4	Digital Input / Output	MCU Port D Bit 4 / Low Leak Wake Up Port 14/ SPI module 1 PCS0 / UART2 Receiver input / TPM module 0 Channel 4	
28	PTD5/ADC0_SE6b/SPI1_SCK/UART2_TX/TPM0_CH5	Digital Input / Output / Analog Input	MCU Port D bit 5 / ADC0 Single Ended analog channel input SE6b / SPI1 clock / UART2 TX / TPM module 0 Channel 5	
29	PTD6/ADC0_SE7b/LLWU_P15/SPI1_MOSI/UART0_RX/SPI1_MISO	Digital Input / Output / Analog Input	MCU Port D bit 6 / ADC0 Single Ended analog channel input SE7b / Low leakage Wake Up port 15 / SPI1 MOSI / UART0 RX / SPI module 1 MISO	
30	NC		No Connect	
31	PTD7/SPI0_MISO/UART0_TX/SPI1_MOSI	Digital Input/ Output	MCU Port D Bit 7 / SPI module 0 MISO / UART module 0 Transmit / SPI module 1 MOSI	
32	PTE0/SPI1_MISO/UART1_TX/RTC_CLKOUT/CMP0_OUT/ I2C1_SDA	Digital Input/ Output	MCU Port E Bit 0 / SPI module 1 MISO / UART module 1 Transmit / Real Time Counter Clock Output / Comparator 0 Analog voltage Output / I2C1 Bus Data	
33	PTA20/RESETB	Digital Input/ Output	MCU Port A Bit 20/MCU $\overline{\text{RESET}}$	
34	PTE1 / SPI1_MOSI / UART1_RX /SPI1_MISO / I2C1_SCL	Digital Input/ Output	MCU Port E Bit 1 / SPI module 1 MOSI / UART module 1 RX / SPI1_MISO / I2C1_SCL	
35	VBAT2 (RF)	Power Input	Transceiver VDD	Connect to system VDD supply
36	GND/SCAN (RF)	Power Input	Transceiver Ground	Connect to ground
37	RXTX (RF)	Digital Output	Transceiver RX / TX RF Switch Control Output; high when in TX	
38	GND_PA2 (RF)	Power Input	Transceiver RF Ground	Connect to ground
39	RFIO (RF)	RF Input / Output	Transceiver RF Input / Output	
40	GND_PA1 (RF)	Power Input	Transceiver RF Ground	Connect to ground
41	PA_BOOST (RF)	RF Output	Transceiver Optional High-Power PA Output	
42	VR_PA (RF)	Power Output	Transceiver regulated output voltage for VR_PA use.	De-coupling cap suggested.
43	VBAT1 (RF)	Power Input	Transceiver VDD for RF circuitry	Connect to system VDD supply

Table 2-1. Pin Function Description (Sheet 4 of 5)

Pin #	Pin Name	Type	Description	Functionality
44	VR_ANA (RF)	Power Output	Transceiver regulated output voltage for analog circuitry.	Decouple to ground with 100 nF capacitor
45	VR_DIG (RF)	Power Output	Transceiver regulated output voltage for digital circuitry.	Decouple to ground with 100 nF capacitor
46	XTA (RF)	Xtal Osc	Transceiver crystal reference oscillator	Connect to 32 MHz crystal and load capacitor
47	XTB (RF)	Xtal Osc	Transceiver crystal reference oscillator	Connect to 32 MHz crystal and load capacitor
48	RESET (RF)	Digital Input	Transceiver hardware reset input	Typically driven from MCU GPIO
49	DIO0/PTE2/SPI1_SCK	Digital Input/ Output	Internally connected to Transceiver GPIO bit 0 and MCU Port E bit 2 / SPI1 clock	MCU IO and Transceiver IO connected onboard
50	DIO1/PTE3/SPI1_MISO/ SPI1_MOSI	Digital Input/ Output	Internally connected to Transceiver GPIO bit 1 and MCU Port E bit 3 /SPI1 in or out	MCU IO and Transceiver IO connected onboard
51	DIO2	Digital Input/ Output	Transceiver GPIO Bit 2	
52	DIO3	Digital Input/ Output	Transceiver GPIO Bit 3	
53	DIO4	Digital Input/ Output	Transceiver GPIO Bit 4	
54	DIO5/CLKOUT	Digital Input/ Output	Transceiver GPIO Bit 5 / ClkOut	Commonly programmed as ClkOut to supply MCU clock; connect to Pin 15 PTA18/EXTAL0.
55	VDD	Power Input	MCU VDD supply	Connect to VDD supply
56	VDDAD	Power Input	MCU Analog supply	Connect to Analog supply
57	MISO/PTC7/SPI0_MISO/ SPI0_MOSI	Digital Input/ Output	Internal SPI data connection from Transceiver MISO bit 1 to MCU SPI0 (Port C bit 7)	• MCU IO and Transceiver IO connected onboard • MCU IO must be configured for this connection • SPI0 is dedicated to radio interface; not for application usage

Table 2-1. Pin Function Description (Sheet 5 of 5)

Pin #	Pin Name	Type	Description	Functionality
58	NSS/PTD0/SPI0_PCS0	Digital Input/ Output	Internal SPI select connection between Transceiver NSS and MCU SPI0 (Port D bit 0)	• MCU IO and Transceiver IO connected onboard • MCU IO must be configured for this connection • SPI0 is dedicated to radio interface; not for application usage
59	SCK/PTC5/SPI0_SCK	Digital Input/ Output	Internal SPI clock connection between Transceiver SCK and MCU SPI0 (port C bit 5)	• MCU IO and Transceiver IO connected onboard • MCU IO must be configured for this connection • SPI0 is dedicated to radio interface; not for application usage
60	MOSI/PTC6/SPI0_MOSI/ SPI0_MISO	Digital Input/ Output	Internal SPI data connection to Transceiver MOSI bit 1 to MCU SPI0 (Port C bit 6)	• MCU IO and Transceiver IO connected onboard • MCU IO must be configured for this connection • SPI0 is dedicated to radio interface; not for application usage
FLAG	VSS	Power input	External package flag. Common VSS	Connect to ground.

2.3 Internal Functional Interconnects

The MCU provides control to the transceiver through the SPI0 Port and receives status from the transceiver from the DIOx pins. Certain interconnects between the devices are routed in the package. In addition, the signals are brought out to external pads for monitoring, but only SPI1 is intended for applications usage. SPI0 is dedicated to the radio interface and should not be used for applications.

Table 2. MKW01Z128 Internal Functional Interconnects

Pin #	MCU Signal	Transceiver Signal	Description
49	DIO0/PTE2/SPI1_ SCK	DIO0	Transceiver DIO0 can be programmed to provide status to the MCU
50	DIO1/PTE3/SPI1_ MISO/SPI1_MOSI	DIO1	Transceiver DIO1 can be programmed to provide status to the MCU

Table 2. MKW01Z128 Internal Functional Interconnects

Pin #	MCU Signal	Transceiver Signal	Description
57	MISO/PTC7/SPI0_ MISO/SPI0_MOSI	MISO	SPI data from transceiver to MCU
58	NSS/PTD0/SPI0_ PCS0	NSS	SPI chip select
59	SCK/PTC5/SPI0_ SCK	SCK	SPI Clock
60	MOSI/PTC6/SPI0_ MOSI/SPI0_MISO	MOSI	SPI data from MCU to transceiver

NOTE

- As shown in Table 2, the MCU SPI Port pin selection must be configured by software by writing the corresponding port multiplex control registers
- The transceiver DIO pins must be programmed to provide desired status

Chapter 3 Signal Multiplexing and Signal Descriptions

3.1 Introduction

To optimize functionality in small packages, pins have several functions available via signal multiplexing. This chapter illustrates which of this device's signals are multiplexed on which external pin.

The Port Control block controls which signal is present on the external pin. Reference that chapter to find which register controls the operation of a specific pin.

3.2 Signal Multiplexing Integration

This section summarizes how the module is integrated into the device. For a comprehensive description of the module itself, see the module's dedicated chapter.

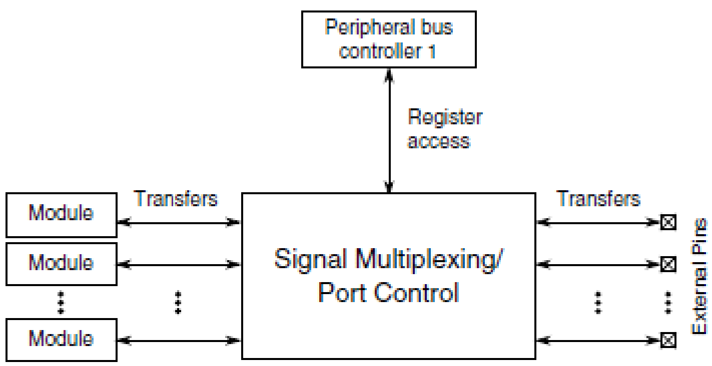


Figure 3-1. Signal Multiplexing Integration

Table 3-1. Reference Links to Related Information

Topic	Related Module	Reference
Full description	Port control	Section 3.2.1, "Port control and interrupt module features"
System memory map		Section 1.7.2, "MCU overview"
Clocking		Clock distribution
Register access	Peripheral bus controller	Peripheral bridge

3.2.1 Port control and interrupt module features

- 32-pin ports

NOTE

Not all pins are available on the device. See the following sections for details.

- Port A and port D are each assigned one interrupt. For DMA requests, port A and port D each have a dedicated input to the DMA multiplex.
- Port A is assigned a dedicated interrupt and port C and port D share an interrupt. For DMA requests, port A, port C, and port D each have a dedicated input to the DMA multiplex.

The reset state and read/write characteristics of the bit fields within the PORTx_PCRn registers are summarized in the table below.

Table 3-2.

This field of PORTx_PCRn	Generally resets to	Except for	Resets to	Configurability
PS	1	PTA0	0	Yes - All GPIO are configurable
PE	0	PTA0 and PTA2	1	Yes - All GPIO are configurable
DSE	0	No exceptions	—	4 pins are configurable for High Drive (PTB0, PTB1, PTD6, PTD7). All others are fixed for Normal Drive and the associated DSE bit is read only.
SRE	1	PTA3, PTA4, PTB17, PTC3, PTC4, PTC5, PTC6, PTC7, PTD4, PTD5, PTD6 PTD7	0	Yes - All GPIO are configurable
MUX	000	PTA0, PTA3, PTA4, PTA20	111	Yes - All GPIO are configurable
PFE	0	No exceptions - all PFE are cleared on reset. [1]	—	The GPIO shared with NMI_b pin is configurable. All other GPIO is fixed and read only.
IRQC	000	No exceptions - all are cleared on reset.	—	Only implemented for ports that support interrupt and DMA functionality.
ISF	0	No exceptions - all are cleared on reset.	—	Only implemented for ports that support interrupt and DMA functionality.

1 The $\overline{\text{RESET}}$ pin has the passive analog filter fixed enabled when functioning as the $\overline{\text{RESET}}$ pin (FOPT[RESET_PIN_CFG] = 1) and fixed disabled when configured for other shared functions.

3.2.2 Clock gating

The clock to the port control module can be gated on and off using the SCGC5[PORTx] bits in the SIM module. These bits are cleared after any reset, which disables the clock to the corresponding module to conserve power. Prior to initializing the corresponding module, set SCGC5[PORTx] in the SIM module to enable the clock. Before turning off the clock, make sure to disable the module. For more details, refer to the clock distribution chapter.

3.2.3 Signal multiplexing contraints

A given peripheral function must be assigned to a maximum of one package pin. Do not program the same function to more than one pin.

To ensure the best signal timing for a given peripheral's interface, choose the pins in closest proximity to each other.

3.3 Pin Assignments and Signal Multiplexing

The following table shows the signals available on each pin and the locations of these pins on the MKW01. The Port Control Module is responsible for selecting which ALT functionality is available on each MCU pin. Both MCU and transceiver pins are shown, for transceiver pin assignment see 6.3.1 DIO Mapping. For those package pins which are connected internally to both the MCU and the transceiver, both devices must be configured in software for the appropriate function. Likewise where an MCU pin is connected to a transceiver pin off-chip.

Table 3-3. MKW01 Pin Assignments and Signal Multiplexing (Sheet 1 of 4)

MKW01 Pin No.	Pin Name	MCU die	XCVR die	Default	alt 0	alt 1	alt 2	alt 3	alt 4	alt 5	alt 6	alt 7
1	VREFH	VREFH		VREFH								
2	VREFL	VREFL		VREFL								
3	VSSA	VSSA		VSSA								
4	VSS1	VSS1		VSS1								
5	PTE16	PTE16		ADC0_DP1/ADC0_SE1	ADC0_DP1/ADC0_SE1	PTE16	SPI0_PCS0	UART2_TX	TPM_CLKIN0			
6	PTE17	PTE17		ADC0_DM1/ADC0_SE5a	ADC0_DM1/ADC0_SE5a	PTE17	SPI0_SCK	UART2_RX	TPM_CLKIN1		LPTMR0_ALT3	
7	PTE18	PTE18		ADC0_DP2/ADC0_SE2	ADC0_DP2/ADC0_SE2	PTE18	SPI0_MOSI		I2C0_SDA	SPI0_MISO		

Table 3-3. MKW01 Pin Assignments and Signal Multiplexing (Sheet 2 of 4)

MKW01 Pin No.	Pin Name	MCU die	XCVR die	Default	alt 0	alt 1	alt 2	alt 3	alt 4	alt 5	alt 6	alt 7
8	PTE19	PTE19		ADC0_DM2/ADC0_SE6a	ADC0_DM2/ADC0_SE6a	PTE19	SPI0_MISO		I2C0_SCL	SPI0_MOSI		
9	PTE30	PTE30		DAC0_OUT/ADC0_SE23/CMP0_IN4	DAC0_OUT/ADC0_SE23/CMP0_IN4	PTE30		TPM0_CH3	TPM_CLKIN1			
10	PTA0	PTA0		SWD_CLK	TSI0_CH1	PTA0	TPM0_CH5	SWD_CLK				
11	PTA3	PTA3		SWD_DIO	TSI0_CH4	PTA3	I2C1_SCL	TPM0_CH0				
12	PTA4	PTA4		NMI_b	TSI0_CH5	PTA4	I2C1_SDA	TPM0_CH1				
13	PTA2	PTA2		DISABLED	TSI0_CH3	PTA2	UART0_TX	TPM2_CH1				
14	PTA1	PTA1		DISABLED	TSI0_CH2	PTA1	UART0_RX	TPM2_CH0				
15	PTA18	PTA18		EXTAL0	EXTAL0	PTA18		UART1_RX	TPM_CLKIN0			
16	PTA19	PTA19		XTAL0	XTAL0	PTA19		UART1_TX	TPM_CLKIN1		LPTMR0_ALT1	
17	PTB0	PTB0		ADC0_SE8/TSI0_CH0	ADC0_SE8/TSI0_CH0	PTB0/LLWU_P5	I2C0_SCL	TPM1_CH0				
18	PTB1	PTB1		ADC0_SE9/TSI0_CH6	ADC0_SE9/TSI0_CH6	PTB1	I2C0_SDA	TPM1_CH1				
19	VDD	VDD		VDD								
20	VSS1	VSS1		VSS1								
21	PTB2	PTB2		ADC0_SE12/TSI0_CH7	ADC0_SE12/TSI0_CH7	PTB2	I2C0_SCL	TPM2_CH0				
22	PTB17	PTB17		TSI0_CH10	TSI0_CH10	PTB17	SPI1_MISO	UART0_TX	TPM_CLKIN1	SPI1_MOSI		
23	PTC4	PTC4		DISABLED		PTC4/LLWU_P8	SPI0_PCS0	UART1_TX	TPM0_CH3	I2S0_MCLK		

Table 3-3. MKW01 Pin Assignments and Signal Multiplexing (Sheet 3 of 4)

MKW01 Pin No.	Pin Name	MCU die	XCVR die	Default	alt 0	alt 1	alt 2	alt 3	alt 4	alt 5	alt 6	alt 7
24	PTC1	PTC1		ADC0_SE15/TSI0_CH14	ADC0_SE15/TSI0_CH14	PTC1/LLWU_P6/RTC_CLKIN	I2C1_SCL		TPM0_CH0		I2S0_TXD0	
25	PTC2	PTC2		ADC0_SE11/TSI0_CH15	ADC0_SE11/TSI0_CH15	PTC2	I2C1_SDA		TPM0_CH1		I2S0_TX_FS	
26	PTC3	PTC3		DISABLED		PTC3/LLWU_P7		UART1_RX	TPM0_CH2	CLKOUT	I2S0_TX_BCLK	
27	PTD4	PTD4		DISABLED		PTD4/LLWU_P14	SPI1_PCS0	UART2_RX	TPM0_CH4			
28	PTD5	PTD5		ADC0_SE6b	ADC0_SE6b	PTD5	SPI1_SCK	UART2_TX	TPM0_CH5			
29	PTD6	PTD6		ADC0_SE7b	ADC0_SE7b	PTD6/LLWU_P15	SPI1_MOSI	UART0_RX		SPI1_MISO		
30	NC	NC										
31	PTD7	PTD7		DISABLED		PTD7	SPI1_MISO	UART0_TX		SPI1_MOSI		
32	PTE0	PTE0		DISABLED		PTE0	SPI1_MISO	UART1_TX	RTC_CLKOUT	CMP0_OUT	I2C1_SDA	
33	PTA20	PTA20		RESET_b		PTA20						RESET_b
34	PTE1	PTE1		DISABLED		PTE1	SPI1_MOSI	UART1_RX		SPI1_MISO	I2C1_SCL	
35	VBAT2		VBAT2	VBAT2								
36	GND/SCAN		GND/SCAN	GND/SCAN								
37	RXTX		RXTX	RXTX								
38	GND_PA2		GND_PA2	GND_PA2								
39	RFIO		RFIO	RFIO								
40	GND_PA1		GND_PA1	GND_PA1								
41	PA_BOOST		PA_BOOST	PA_BOOST								
42	VR_PA		VR_PA	VR_PA								
43	VBAT1		VBAT1	VBAT1								
44	VR_ANA		VR_ANA	VR_ANA								

Table 3-3. MKW01 Pin Assignments and Signal Multiplexing (Sheet 4 of 4)

MKW01 Pin No.	Pin Name	MCU die	XCVR die	Default	alt 0	alt 1	alt 2	alt 3	alt 4	alt 5	alt 6	alt 7
45	VR_DIG		VR_DIG	VR_DIG								
46	XTA		XTA	XTA								
47	XTB		XTB	XTB								
48	RESET		RESET	RESET								
49	DIO0 / PTE2	PTE2	DIO0	DISABLED	[1]	PTE2	SPI1_SCK					
50	DIO1 / PTE3	PTE3	DIO1	DISABLED	[1]	PTE3	SPI1_MISO			SPI1_MOSI		
51	DIO2		DIO2	[1]								
52	DIO3		DIO3	[1]								
53	DIO4		DIO4	[1]								
54	DIO5		DIO5	CLKOUT	[1]							
55	VDD	VDD		VDD								
56	VDDA	VDDA		VDDA								
57	MISO / PTC7	PTC7	MISO	CMP0_IN1	CMP0_IN1	PTC7	SPI0_MISO		I2S0_RX_FS	SPI0_MOSI		
58	NSS / PTD0	PTD0	NSS	DISABLED		PTD0	SPI0_PCS0		TPM0_CH0			
59	SCK / PTC5	PTC5	SCK	DISABLED		PTC5/LLWU_P9	SPI0_SCK	LPTMR0_ALT2	I2S0_RXD0		CMP0_OUT	
60	MOSI / PTC6	PTC6	MOSI	CMP0_IN0	CMP0_IN0	PTC6/LLWU_P10	SPI0_MOSI	EXTRG_IN	I2S0_RX_BCLK	SPI0_MISO	I2S0_MCLK	

[1] See Section 7.3, “Digital IO Pins Mapping”, for DIO mapping.

Chapter 4
System Considerations

4.1 Introduction

The MKW01Z128 is the embodiment of a sub-1 GHz wireless node in a single SiP package. All control of the node is done through the in-package Kinetis KL26 48 MHz processor, and all MCU peripherals, MCU GPIO, transceiver functionality, and transceiver GPIO are manipulated by the processor. The MCU GPIO and MCU peripherals are accessed as ports from the MCU internal bus and can be programmed directly.

Communication to the transceiver is through the common SPI0 bus and several MCU GPIO lines. Primary interface with the transceiver is through the SPI command structure that allows reading/writing registers and provides initialization of parameters, reading of status, and control of transceiver operation. The transceiver also has two status signals tied to MCU GPIO internally and requires several others to be tied externally.

This chapter presents information addressing application and operation of the node from a system level. The areas considered here are also covered in greater detail in the following sections of the book. The book is organized such that the first three chapters present the top-level view of the MKW01Z128 device and the following chapters present individual functions in detailed descriptions.

4.2 Power connections

The MKW01Z128 power connections at the SiP level are listed in Table 4-1.

Table 4-1. Power pin descriptions

Pin #	Pin Name	Type	Description	Functionality
3	VSSA	Power Input	MCU ADC Ground	Connect to ground
4	VSS	Power Input	MCU Ground	Connect to ground
19	VDD	Power Input	MCU VDD	Connect to MKW01Z128 VDD supply
20	VSS	Power Input	MCU Ground	Connect to ground
35	VBAT2	Power Input	Transceiver VDD	Connect to MKW01Z128 VDD supply
36	GND/SCAN	Power Input	Transceiver Ground	Connect to ground
38	GND_PA2 (RF)	Power Input	Transceiver RF Ground	Connect to ground
40	GND_PA1 (RF)	Power Input	Transceiver RF Ground	Connect to ground

Table 4-1. Power pin descriptions (continued)

Pin #	Pin Name	Type	Description	Functionality
42	VR_PA	Power Output	Transceiver regulated output voltage for VR_PA use.	
43	VBAT1 (RF)	Power Input	Transceiver VDD for RF circuitry	Connect to MKW01Z128 VDD supply
44	VR_ANA	Power Output	Transceiver regulated output voltage for analog circuitry.	Decouple to ground with 100 nF capacitor
45	VR_DIG	Power Output	Transceiver regulated output voltage for digital circuitry.	Decouple to ground with 100 nF capacitor
55	VDD	Power Input	MCU VDD supply	Connect to MKW01Z128 VDD supply
56	VDDA	Power Input	MCU ADC VDD	Connect to MKW01Z128 VDD supply
FLAG[1]	VSS	Power input	External package flag. Common VSS	Connect to ground.

[1] Flags on bottom of package are electrically separate. Both must be connected to ground.

When designing power to the MKW01Z128 SiP, the following points need to be considered:

- The SiP package has two ground flags (VSS) on chip the package (pin 4 and pin 20). These are separated and both must be connected to system ground.
- The MCU VDD power supply connections include
  - — VDD (Pin 19)
  - — VDD (Pin 55)
  - — The VDDA (Pin 56) analog supply to the MCU ADC is also normally wired to the common source supply.
- For the transceiver the primary power inputs include
  - — VBAT2 (Pin 35)
  - — VBAT1 (Pin 43)
  - — Both VBAT1 and VBAT2 should be powered together from the same circuitry
- The transceiver provides on-chip voltage regulator outputs for bypassing -
  - — VR_ANA (Pin 44) regulated voltage to analog circuitry; bypass to ground
  - — VR_DIG (Pin 45) regulated voltage to digital circuitry; bypass to ground
- The transceiver provides a regulated output VR_PA (Pin 42) for with RF power boost mode.
- Additional system ground pins include -
  - — VSS (Pin 4 and Pin 20) - MCU ground
  - — VSSA (Pin 3) - MCU ATD ground
  - — GND (Pin 36) - transceiver ground
  - — GND_PA1 (Pin 40) and GND_PA2 (Pin 38) - transceiver RF grounds

Power supply connections are shown in Figure 4-1.

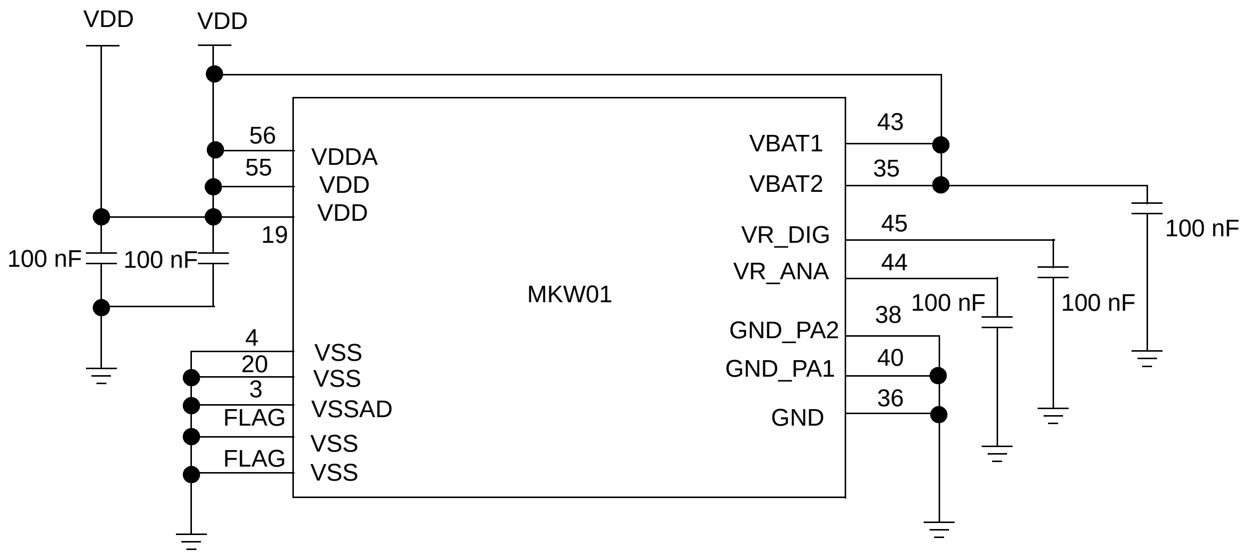


Figure 4-1. MKW01Z128 power supply connections

NOTE

Depending on the application, the VREFH high reference voltage for the ADC module is commonly also tied to the VDD common supply and the VREFL low reference voltage is tied to ground.

4.3 System functional interconnects

The MKW01Z128 comprises two separate devices in a single package. The MCU controls the transceiver and there are connections between the devices for several functions.

- Some connections are provided on chip
- Additional external connections may also be used depending on the application needs.
  - — Transceiver reset
  - — Clock interconnect
  - — Additional transceiver status
  - — Enhanced packet performance

4.3.1 In-package Connections (SPI Channel and Status)

The internal (in-package) device connections are listed in Table 2. These include:

- SPI communication channel - see Chapter 8, “MKW01Z128 Transceiver - MCU SPI Interface”
- Transceiver DIO0 and DIO1 outputs as status - these status outputs are used to manage the data
- flow and transceiver sequencer during radio packet mode operation. Enhanced performance can be obtained by additionally connecting these pins externally to other GPIO pins described below.

4.3.2 System Reset

The MKW01Z128 system does not have a master input that resets the entire SiP:

- The MCU reset input is pin 33, RESET_b — use of this pin as an external reset is optional, and the MCU has a separate power-on reset (POR).
- The transceiver has an independent reset pin (RESET), pin 48 — this signal is typically connected to an MCU GPIO to provide total software control of the transceiver

NOTE

It is recommended that the MCU be connected to the transceiver reset via an MCU GPIO pin to provide best overall hardware control.

4.3.2.1 MCU Reset pin (pin 33)

On this device, $\overline{\text{RESET}}$ is a dedicated input pin for which filtering can be enabled. This pin is open drain and has an internal pullup device. This pin can be disabled or configured as a GPIO (PTA20). Asserting $\overline{\text{RESET}}$ (low) wakes the device from any mode. During a pin reset, the RCM's SRS0 PIN bit is set. The MCU's Low Voltage Detect (LVD) can also generate a reset.

4.3.2.2 Transceiver Reset

The transceiver can be reset via two means:

- A power-on reset (POR) of the MKW01Z128 transceiver is triggered when VDD is applied to VBAT1 and VBAT2.
- A hardware reset can be issued by controlling Pin 48 (transceiver RESET).

4.3.2.2.1 Transceiver POR

Similar to the MCU, a transceiver POR is internally generated when power is applied to VDD. See Figure 4-2 for the transceiver POR timing diagram.

- The transceiver hardware RESET pin is bidirectional and is first driven to high as the result of the POR.
- The RESET signal is then driven low, and the application must wait for 10 ms from the end of the POR cycle before commencing communications over the SPI bus.
- RESET should be left floating during the POR sequence.

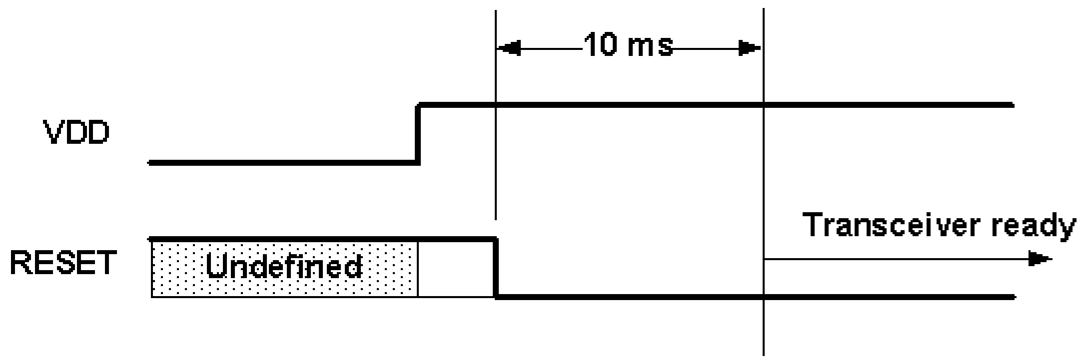


Figure 4-2. POR Timing Diagram

NOTE

Any CLKOUT activity can also be used to detect when the chip is ready.

4.3.2.2.2 Transceiver Hardware Reset

A hardware reset of the transceiver on MKW01Z128 is also possible by asserting RESET high for a minimum of one hundred microseconds, and then releasing. The application must wait 5 ms before using the chip.

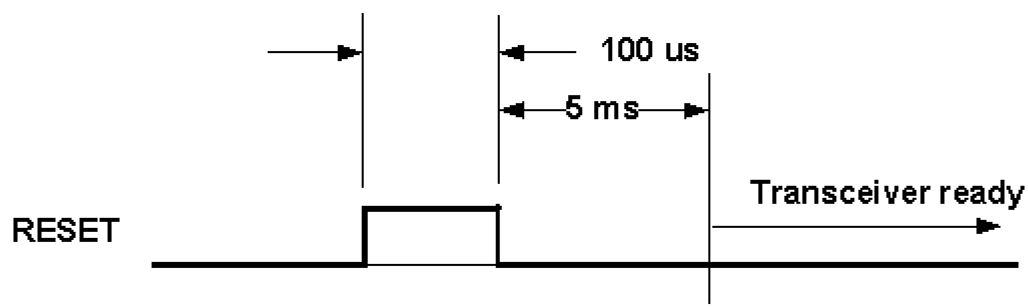


Figure 4-3. Manual Reset Timing Diagram

NOTE

While RESET is driven high, additional current consumption of up to ten milliamps may be seen on VDD.

4.3.2.3 MCU Control of Transceiver Reset

It is recommended to provide hardware reset capability of the transceiver via an MCU GPIO externally connected to the transceiver RESET pin. For Freescale applications software, MCU signal PTE30 is the preferred GPIO to control the RESET.

4.3.3 External Clock Connections

It is possible that the transceiver can supply a clock source to the MCU through external connections. The transceiver can output a clock signal on pin 54 DIO5/CLKOUT. The transceiver can be configured (via the SPI) to select various clock frequencies that are divided from the transceiver external crystal frequency being applied to XTA and XTB pins. This output clock can be connected to pin 15 PTA18/EXTAL0 , externally, and provide the clock source to the MCU's MCG module, thus providing the CPU and System clock of the MCU.

NOTE

From POR and from an MCU reset (on pin 33) the MCU will by default select the internal RC oscillator as the clock source to the CPU.

4.4 System Clock Sources and Configurations

The MKW01Z128 clock connections are shown in Figure 4-4. The device allows for a wide array of system clock configurations.

- Pins are provided for a separate external clock source for the CPU. The external clock source can by derived from a crystal oscillator or from an external clock source
- The transceiver optionally provides a ClkOut programmable frequency clock output that can be used as an external source to the CPU. As a result, a single crystal system clock solution is possible. This is the preferred configuration
- Pins are provided for a 32 MHz crystal for the transceiver reference oscillator (crystal can be 28 to 33 MHz but all frequencies derived from Fref must be appropriately calculated)
- The MCU contains an internal nominal 32 kHz clock oscillator (which can be trimmed) that can be used to run the MCU
- Out of reset, the MCU uses the internal oscillator and the on-chip FLL to generate an approximately 20 MHz clock for start-up. This allows recovery from stop or reset without a long crystal start-up delay

In addition, the transceiver has an on-chip RC oscillator that is used only internally for triggering periodic listen modes.

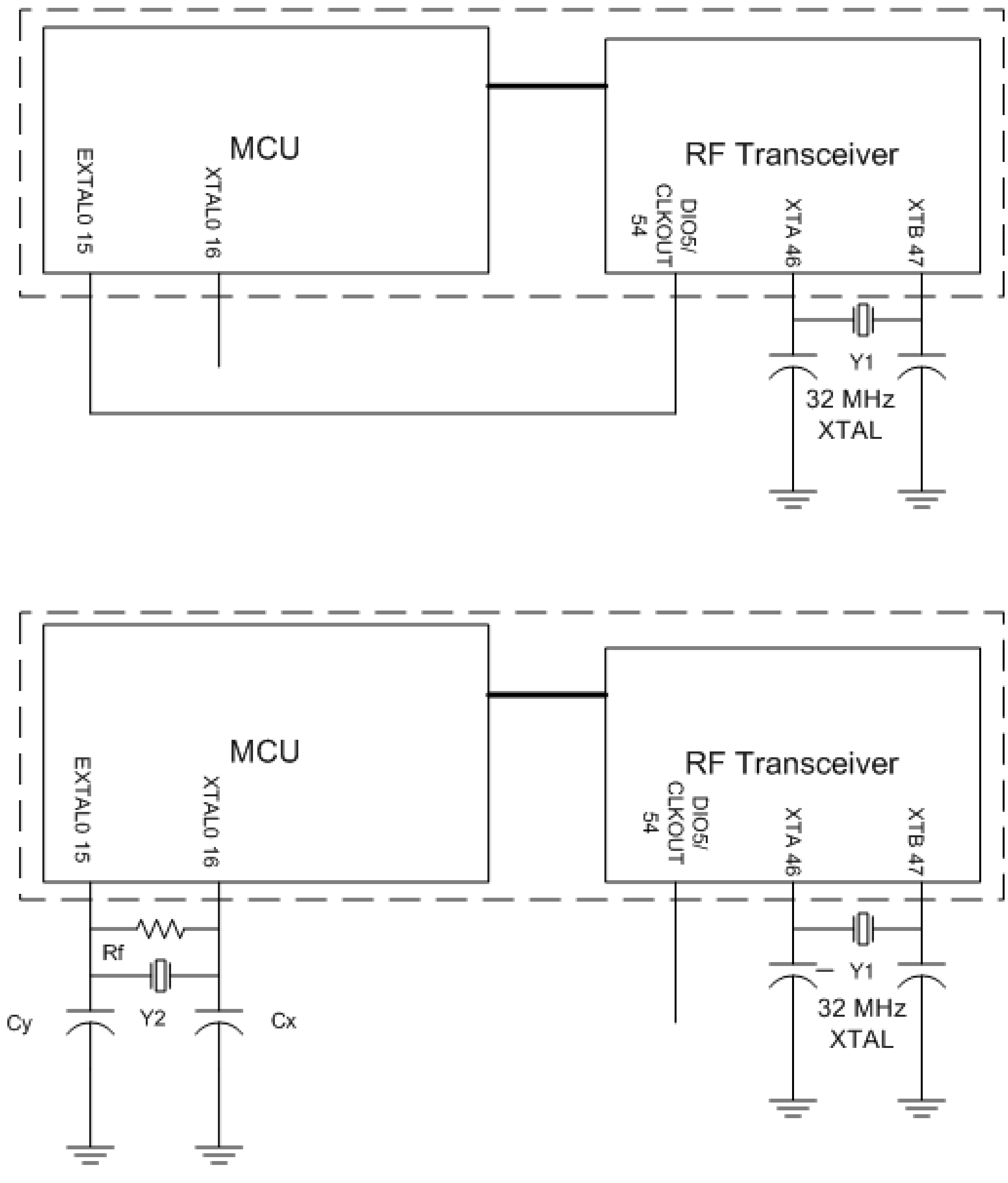


Figure 4-4. MKW01Z128 Clock Connections (Upper figure is the preferred configuration.)

4.4.1 Additional Transceiver Status Signals

The MKW01Z128 transceiver has a total of six outputs (DIO5:DIO0) that can be programmed as status indicators:

- DIO1 and DIO0 are connected to MCU GPIOs, PTE3 & PTE2, internally to the package. For certain software configurations, improved performance can be obtained by connecting to additional GPIOs, PTC4, and PTC3, off chip. These connections are used by applications software in packet mode.

- At the user's discretion, the additional DIO5:DIO2 can be connected externally to the MCU GPIO and programmed as status indicators -
  — Use of selected signals must be programmed on the transceiver
  — If interrupt request (IRQ) capability is desired, any status signal must be connected to an IRQ capable GPIO pin.

4.4.2 Transceiver Oscillator

The transceiver crystal oscillator is the main timing reference of the device. The transceiver oscillator source must always be present and an external crystal is typically used to implement the oscillator, although an external TCXO may also be used (see Section 5.5.1, "Reference Oscillator"). The source frequency is normally 32 MHz.

In Figure 4-4 crystal Y1 and two capacitors form the transceiver crystal oscillator circuit. An off board feedback resistor between input XTA and output XTB is not required. An important parameter for the crystal Y1 is the load capacitance. The oscillator needs to see a balanced load capacitance at each terminal, and as a result, the sum of the stray capacitance of the pcb board, device pin (XTA or XTB), and load capacitor at each terminal should be equal. The amount of external load capacitance is determined by the specific crystal specification.

NOTE

- There is no on-chip trim capacitance, therefore the user should evaluate and size the external load capacitors to center the oscillator frequency within the cut tolerance of the crystal.
- The frequency accuracy of the crystal (cut tolerance plus temperature variation) must be matched to the required specification of the application.
- To compensate for reference error, actual RF frequency can be adjusted to the desired frequency by offsetting of the PLL control word, Frf.

4.4.2.1 Crystal Resonator Specification

Table 4-2 shows the crystal resonator specification for the crystal reference oscillator circuit of the MKW01Z128 transceiver. This specification covers the full range of operation and is employed in the reference design.

Table 4-2. Crystal Specification

Symbol	Description	Conditions	Min	Typ	Max	Unit
FXOSC	XTAL Frequency		26	32	33	MHz
Rs	XTAL Serial Resistance		-	30	140	ohms
C0	XTAL Shunt Capacitance		-	2.8	7	pF
CLOAD	External Foot Capacitance	On each pin XTA and XTB	8	16	22	pF

NOTE

- The initial frequency tolerance (cut tolerance), temperature stability and aging performance should be chosen in accordance with the target operating temperature range and the receiver bandwidth selected.
- The loading capacitance should be applied externally, and adapted to the actual C_{load} specification of the crystal.
- A minimum crystal frequency of 28 MHz is required to cover the 863-870 MHz band and 29 MHz for the 902-928 MHz band.

4.4.2.2 Transceiver ClkOut Output (DIO5)

The reference frequency, or a fraction of it, can be provided as an output on DIO5. Use of the ClkOut output allows:

- Driving the clock source to the MCU -
  — Saves the cost of an additional crystal.
  — ClkOut can be made available in any operation mode except Sleep mode and is automatically enabled at power-on reset.
- Trimming of the reference oscillator frequency - ClkOut can be provided as a test point for a frequency counter to allow trimming of the external load capacitance during design/evaluation.

The ClkOut functionality is controlled by programming transceiver Register RegDioMapping2 (0x26) (see Section 7.10, “IRQ and Pin Mapping Registers”):

- Bits 5:4 - control the function of DIO5 and enable ClkOut. See Table 7-2 and Table 7-3.
- Bits 2:0 - control the ClkOut frequency.

Table 4-3 lists ClkOut frequency versus Bits 2:0 setting using a 32 MHz reference source.

Table 4-3. ClkOut Frequency Using 32 MHz Reference Oscillator

ClkOut Bits 2:0	Divide Ratio	ClkOut Frequency (MHz)
000	1	32
001	1/2	16
010	1/4	8
011	1/8	4
100	1/16	2
101	1/32	1
110	**RC**	**62.5 kHz (Automatically enabled)**
111	-	OFF

4.4.3 MCU Clock Sources

The MCU has several options for its primary clock source depending on its mode of operation as well as its hardware configuration**.**

- The ICS module has an on-chip 32 kHz (nominal) oscillator and FLL -
  - Provides start-up run clock
  - The oscillator can be used with or without the FLL
  - The FLL can be used with an external clock/crystal
- External pins are provided to accommodate an external crystal, resonator, or clock source.
- A separate low power 1 kHz oscillator (LPO) can be used for the real time counter (RTC) or the COP timer.

4.4.3.1 MCU External Clock Source

As shown in Figure 4-4, the external pins associated with the MCU clock source are output XTAL and input EXTAL. (External pin names are XTAL0 and EXTAL0.) An external clock source (such as ClkOut) can have a frequency as high as 40 MHz with no minimum frequency. The external source must have compatible logic levels and drive input EXTAL. Note that no external components are required for the clock oscillator if an external source is used.

4.4.3.2 MCU External Crystal Oscillator

Referring again to Figure 4-4, another choice is the use of an external crystal or ceramic resonator, shown as Y2. The MCU oscillator is a Pierce type that can accommodate crystals or resonators in any of four modes:

- 30 to 40 kHz low frequency range crystal — low power
- 30 to 40 kHz low frequency range crystal — high gain
- 3~32 MHz high frequency range crystal — low power
- 3~32 MHz high frequency range crystal — High gain

R_F must be a low-inductance resistor such as carbon composition. Wire-wound resistors, and some metal film resistors, have too much inductance. Cy and Cx normally must be high-quality ceramic capacitors specifically designed for high-frequency applications.

R_F is used to provide a bias path to keep the EXTAL input in its linear range during crystal startup; its value is not generally critical. Typical systems use 1 MΩ to 10 MΩ. Higher values are sensitive to humidity and lower values reduce gain and (in extreme cases) could prevent startup. With the low-power mode, the oscillator has the internal feedback resistor R_F. Therefore, the feedback resistor must not be externally

Cy and Cx are typically not used when a low frequency crystal is chosen. With a high frequency range crystal, these capacitors will be in the 5 pF to 25 pF range and are chosen to match the requirements of a specific crystal or resonator. Take into account printed circuit board (PCB) capacitance and MCU pin capacitance when selecting Cy and Cx. The crystal manufacturer typically specifies a load capacitance which is the series combination of Cy and Cx (which are usually the same size). As a first-order approximation, use 10 pF as an estimate of combined pin and PCB capacitance for each oscillator pin (EXTAL and XTAL).

When using the oscillator in low frequency range modes, the external components Cx and Cy are not required.

When using the oscillator in low power modes, the external R_F is not required and should not be used.

4.4.3.3 MCU Internal Clock Source

The MKW01 has an internal reference clock (nominally 32 kHz) that is used in a number of ways:

- Default MCU clock out of reset - the internal ICS clock module defaults to the internal oscillator enabled and the FLL engaged. The resulting default CPU clock frequency is a nominal 20 MHz (10 MHz bus clock).
- Programmable system clock - either used with or without the FLL, the internal oscillator can remain as the normal RUN frequency source
  - Nominal maximum CPU clock of 48 MHz
  - Total trimmed frequency deviation of +/-2% maximum
- Wake-up clock for low power modes - the internal oscillator can be enabled to clock the RTC to provide a wake-up timer from Stop2 or Stop3

4.4.3.4 LPO 1 kHz Oscillator

The LPO is independent of the ICS module and can be used to clock the RTC or COP. Its period can vary greatly from 0.7–1.3 ms.

4.4.4 System Clock Configurations

Because of the multiple clock configurations of the MCU, the availability of external clock source pins of both the transceiver and the MCU, and the ClkOut output from the transceiver, there are a number of variations for MKW01Z128 system clock configurations. Key considerations for any system clock configuration are:

- The transceiver 32 MHz source (typically the reference crystal oscillator) must always be present.
- Battery-operated application requirements for low power can impact the choices for MCU clock source.
- The system clock configuration will impact system initialization procedures.
- Software requirements can impact MCU processor and bus speed - The user must be aware of the performance requirements for the MCU. The CPU clock is always 2X the internal bus speed, and the application software may impact the required system clock rate.

As far as external connections are concerned, there are three possibilities which are covered in the following sections.

NOTE

In the following sub-sections, it is assumed that the MCU GPIO is connected to drive the transceiver reset.

4.4.4.1 Single crystal with ClkOut driving MCU EXTAL input

The single crystal (transceiver) with ClkOut driving the MCU EXTAL input (external clock) is the most common configuration for low cost and excellent frequency accuracy. The ClkOut frequency is programmable but setting the divide ratio to 1 resulting in a 32 MHz CLKOUT (with a 32 MHz crystal) is recommended to drive the MCU external source.

In this configuration, clock start-up from a reset condition involves:

- MCU reset is released and MCU starts on internal 20 MHz clock, which is derived internally via the FLL from the slow IRC 32 kHz clock.
- Initialization software should reset and then release reset to the transceiver (MCU still running on start-up clock)
- Wait for transceiver start-up.
- Program ClkOut to desired frequency (clkout to 000, divide by 1, recommended)
- Wait for the ClkOut source plus PLL to lock, and then switch MCU clock to external source

If the transceiver is forced to a low power condition, the MCU can revert to the internal oscillator and FLL.

4.4.4.2 Single Crystal with MCU Using Internal Clock Only

The single crystal (transceiver) with the MCU using internal clock only has no real advantage over using the ClkOut output, except for slightly lower power.

In this configuration, clock start-up from a reset condition involves:

- MCU reset is released and MCU starts on internal 20 MHz clock, which is derived internally via the FLL from the slow IRC 32 kHz clock
- Initialization software should assert reset and then release reset to the transceiver (MCU still running on start-up clock)
- Program MCG to desired clock rate if the default is not the preferred choice
- Program transceiver to disable ClkOut.

The MCU does not have a high accuracy time base when using the internal reference.

4.4.4.3 Dual Crystal Operation

The transceiver crystal can be augmented by the use of a second crystal on the MCU. The typical application would use a 32.768 kHz crystal that would allow an accurate time base in the MCU for long power down delays. The obvious disadvantage of this configuration is additional cost.

In this configuration, clock start-up from a reset condition involves:

- MCU reset is released and MCU starts on internal 20 MHz clock, which is derived internally via the FLL from the slow IRC 32 kHz clock

- Initialization software should assert reset and then release reset to the transceiver (MCU still running on start-up clock)
- The MCU clock can be switched to the OSC
- Program transceiver to disable ClkOut

The second crystal is justified when accurate power down time periods are required. The external clock with the 32.768 kHz crystal allows an accurate time tick for the RTC at very low power.

4.4.5 Debug Port Pin Descriptions

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

Table 4-4. Serial wire debug pin description

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

4.5 MKW01Z128 GPIO (Mixed I/O from Transceiver and MCU)

The MKW01Z128 SiP supports a total of 43 GPIO pins that originate from the transceiver and/or the MCU:

- The transceiver provides six pins (DIO5-DIO0) that can be programmed as status -
  — Use of the DIOX are programmed as listed in Table 7-2 and Table 7-3
  — DIO1 and DIO0 are connected internally to MCU GPIOs, PTE3, PTE2, as well as routed to package pins. Enhanced performance can be achieved by routing them externally to other GPIO pins, PTC4 and PTC3.
  — Freescale software commonly uses DIO4 externally connected to PTD4.
  — DIO5 is also ClkOUT and most commonly externally connected to EXTAL0 for the MCU clock source
  — DIO2 often used for DATA out, in conjuction with DIO1 DCLK.
  — DIO3 is uncommitted.
- Four-signal SPI Bus port connected internally and reserved for MCU and transceiver use. On bottom pins 57–60. (PTC7, PTD0, PTC5, PTC6)
- PTE30 is typically used to drive the RESET on the transceiver.
- PTE0, PTE3, are typically used for SWD
- PTA1 and PTA2 are used for UART0 RXD & TXD.
- PTC1 and PTC2 are typically used for I2C.
- PTE17 is used as a software PN output for testing in some Freescale applications.

- — If a crystal is used with the MCU, PTA19/XTAL0 and PTA18/EXTAL0 pins cannot be used as GPIO. These are available only using the internal clock reference.
- — If ClkOut is used to drive the external clock source to the MCU, EXTAL (PTA18) is not available as GPIO.
- — PTA20 can be used as MCU reset.

- fourteen additional MCU GPIO are available for use in the SiP package. Four of them (PTB0, PTB1, PTD6, PTD7) have 20mA drive capability and can drive LEDs directly.

NOTE

Unused GPIO pins should be disabled and if possible left unconnected.

NOTE

Four additional MCU GPIO are not pinned-out and cannot be used. However, they must be initialized for low power operation.

- — If a crystal is used with the MCU, PTA19/XTAL0 and PTA18/EXTAL0 pins cannot be used as GPIO. These are available only using the internal clock reference.
- — If ClkOut is used to drive the external clock source to the MCU, EXTAL0 is not available as GPIO.
- — PTA0, PTA3 and PTA20 are not commonly used as GPIOs because they are dedicated to the MCU debug port (SWD).

4.5.1 MCU GPIO Characteristics

The internal MCU GPIO hardware consists of 5 ports with 32 signals per port for a total of 160 signals (not all are available on the MKW01Z128). There are 8 signals from PTA, 5 from PTB, 9 from PTC, 5 from PTD and 9 from PTE, many of which are dedicated to some function. This can be seen in Figure 1-1.

NOTE

To avoid extra current drain, all unused signals should be disabled. This includes signals not pinned-out on the package.

- The GPIO ports share the pins with other modules and selection of either GPIO or a selection of altenative modules signals are configured via the PORT control module. The PORT control module provides up to 7 possible options for each external pin, and also provides optional pull-up/pull-down devices when GPIO pin is input, and interrupt capability where available. See section 7 of the MCU portion of this reference manual, Port Control.
- For information about how and when on-chip peripheral systems use these pins, refer to the appropriate peripheral chapter of this document

4.5.2 Transceiver DIOX Characteristics

The transceiver status/GPIO consists of 6 signals total (DIO5-DIO0). DIO2 is bidirectional and all others are output only. Immediately after reset, all the DIO are configured as outputs. Use of the DIO is controlled by transceiver registers RegDioMapping1 and RegDioMapping2, see Section 7.10, “IRQ and Pin Mapping Registers”.

4.6 Transceiver RF Configurations and External Connections

The MKW01Z128 transceiver radio has features that allow for a flexible as well as low cost RF interface:

- Sensitivity down to -120 dBm at 1.2 kbps
- High selectivity w/16-tap FIR channel filter
- Robust receiver front end:
  - IIP3 = -18 dBm
  - IIP2 = +35 dBm
  - 80 dB blocking immunity
  - No image frequency response
- Programmable Pout from -18 to +17 dBm in 1 dB steps
- FSK bit rates up to 600 kbps
- FSK, GFSK, MSK, GMSK, and OOK modulators
- Two TX output power configurations

4.6.1 RF Interface Pins

The MKW01Z128 transceiver has the following pins associated with the RF interface:

- VR_PA - regulated voltage supply to the internal PA circuit, to be routed through the matching network
- PA_BOOST - optional secondary high-power TX PA
- RFIO - RF bidirectional input/output; used as input only for PA boost mode
- RXTX - digital output to control RF switch
- GND_PA1- RF ground
- GND_PA2 - RF ground

NOTE

The following descriptions for RF operation are not meant as complete applications information, but rather general information. Freescale supplies evaluation boards as well as complete reference designs, and the user is directed to these designs upon which to build target systems.

There are two basic RF configurations as described in the following sections.

4.6.2 Standard Output Power RF Configuration (Single, Bidirectional Port)

The standard RF configuration for the transceiver is a single, bidirectional port mode (shown in Figure 4-5):

- Only the bidirectional RF pin RFIO is used
- Maximum output power of typically +13 dBm into a 50 ohm load.
- The output power is programmable in approximately 1 dB steps.

- Inductor L8 and capacitor C19 provide an ac blocking network for the PA power supply.
- The remaining external components are a generalized RF network -
  — Provides impedance match between the antenna and the transceiver RFIO
  — Supports an elliptic-function low pass filter to provide TX harmonic trapping and out-of-band suppression
  — The topology for the RF matching network can be used over the various bands of interest with changes in component values. Not all indicated components are used at all frequencies

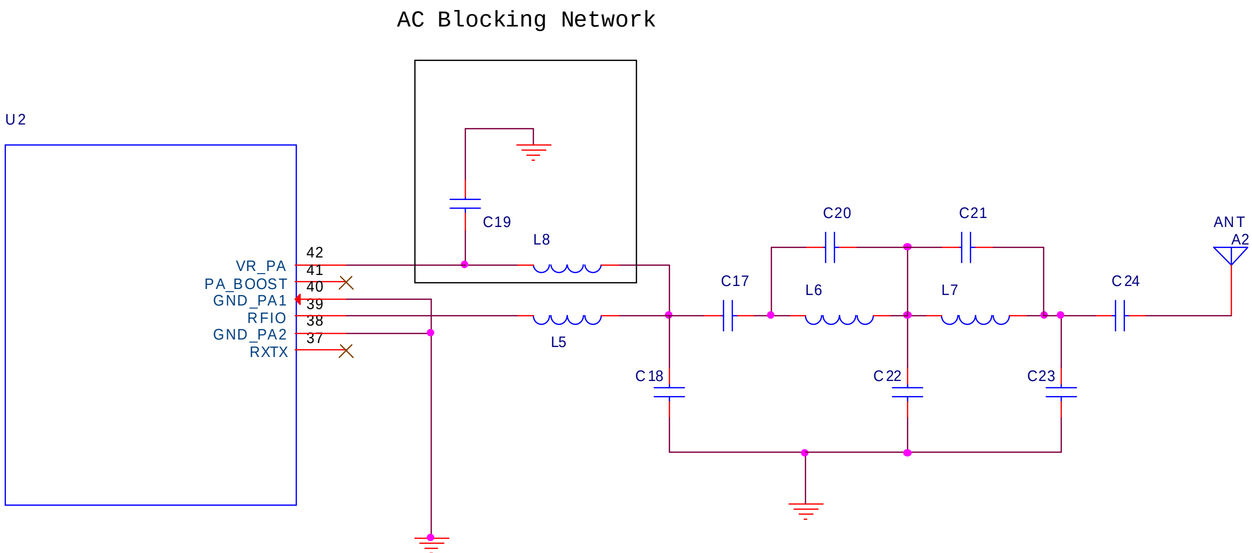


Figure 4-5. Single Port RF Schematic (+13dBm)

4.6.3 Higher Output Power RF Configuration (Dual Port with Optional External Power Amplifier)

The secondary RF configuration for the transceiver is a dual port mode (shown in Figure 4-6):

- Secondary PA output (PA_BOOST) is used as the TX port and the standard RFIO pin is used as the RX port.
- Maximum output power from the PA_BOOST output is typically +17 dBm into a 50 ohm load.
- Inductor L1 and capacitor C1 provide an ac blocking network for the PA_BOOST power supply.
- An external RF switch is required to enable the appropriate path to the antenna -
  — The TX port (PA_BOOST network) is selected for transmit
  — The RFIO port network is selected for receive
  — Device control output RXTX can used to control the antenna switch
- An optional external power amplifier can be inserted between the device TX output and the antenna switch to increase transmitted power up to typically about 1 W maximum.

- Similar to the single-port configuration, generalized RF circuit topologies are shown for both the TX and RX paths -
  - — The TX path network provides both impedance matching and low pass filtering for TX harmonic trapping and out-of-band suppression
  - — The RX path network typically provides only impedance match between the antenna and the transceiver RFIO and can normally be greatly simplified
  - — The topology for the RF matching networks can be used over the various bands of interest with changes in component values. Not all indicated components are used at all frequencies.

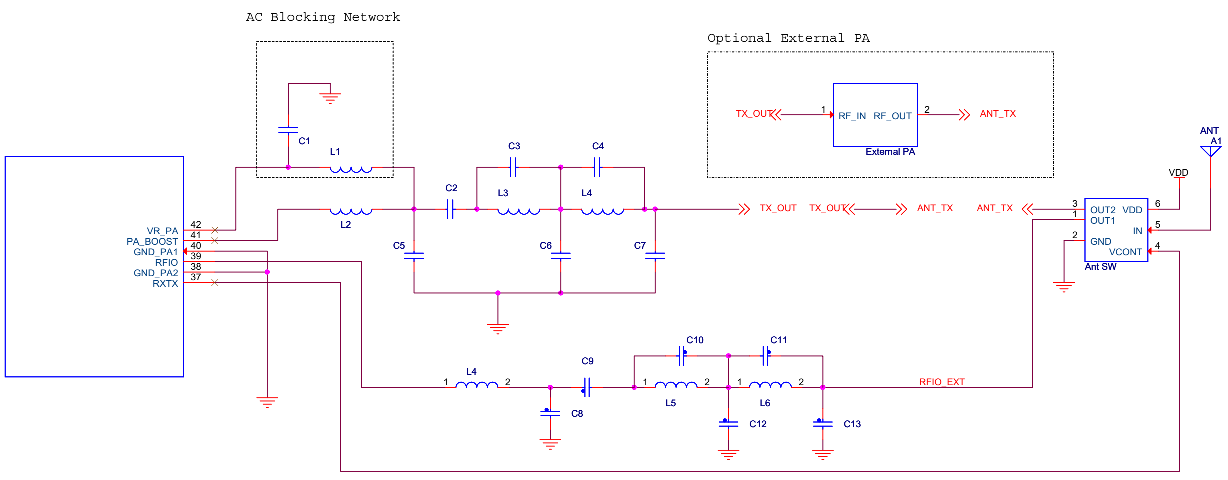


Figure 4-6. Dual Port RF Schematic (+17dBm to 1W)

4.6.4 Filter and Matching Network Component Values

The generalized filter/matching network shown in Figure 4-5 and Figure 4-6 must be evaluated and tuned to its application use and frequency:

- Impedance matching is always required
- Only a transmission path typically uses any low pass filtering elements
- Not all indicated components are used at all frequencies

To provide an initial design configuration for several popular frequency bands, Table 4-5 lists component values versus frequency and use for the generalized component topology of Figure 4-7. For each frequency, the filter components for the PA_BOOST output filter (Dual Port TX) and the Single Port (TX/RX) are given.

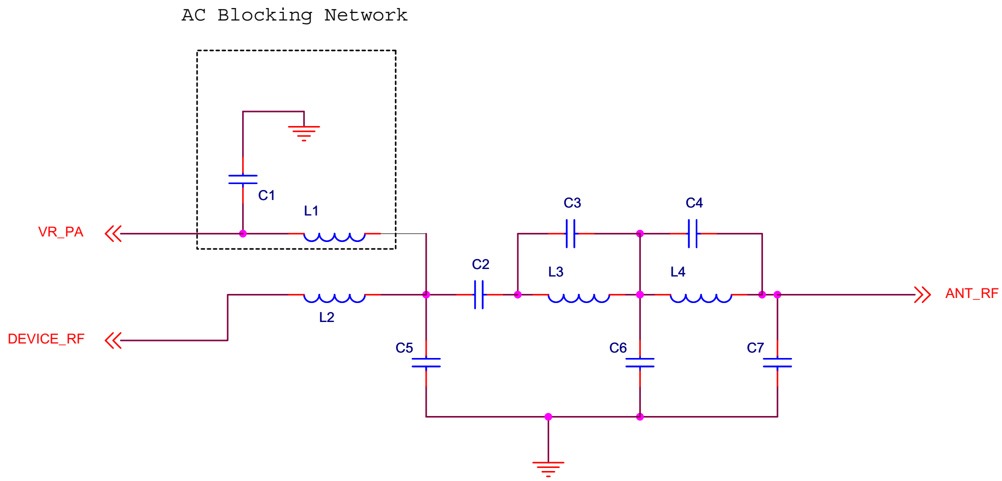


Figure 4-7. General RF Filter/Matching Network Topology

NOTE

These component values are given only as suggested initial design values. The user must evaluate his particular design and adjust/change components as required to meet targeted specifications.

Table 4-5. RF Component Values vs. Frequency Band

Component Designation	434 or 490 MHz		868 MHz or 915 MHz	
	Dual Port TX	Single Port TX/RX	Dual Port TX	Single Port TX/RX
L1	22 nH	33 nH	33 nH	33 nH
L2	Short	Short	2 nH	4.7 nH
L3	12 nH	12 nH	5.6 nH	6.8 nH
L4	12 nH	10 nH	5.6 nH	6.8 nH
C2	10 pF	10 pF	47 pF	6.8 pF
C3	2.4 pF	2.4 pF	dnp or open	dnp or open
C4	dnp	dnp	dnp or open	dnp or open
C5	15 pF	15 pF	6.8 pF	2 pF
C6	15 pF	15 pF	6.8 pF	7.5 pF
C7	8.2 pF	8.2 pF	3.3 pF	5.6 pF

Chapter 5
Sub 1 GHz Transceiver Architecture Description

This chapter describes the architecture and operation of the MKW01Z128 low-power, highly integrated transceiver chip.

5.1 Overview

The MKW01Z128 transceiver is a single-chip integrated circuit ideally suited for today's high performance ISM band RF applications. The transceivers's advanced features set, including state of the art packet engine greatly simplifies system design while the high level of integration reduces the external BOM to a handful of passive decoupling and matching components. It is intended for use as high-performance, low-cost FSK and OOK RF transceiver for robust frequency agile, half-duplex bidirectional RF links, and where stable and constant RF performance is required over the full operating voltage range of the device.

The transceiver is intended for applications over a wide frequency range, including the 433 MHz and 868 MHz European and the 902-928 MHz North American ISM bands. Coupled with a link budget in excess of 135 dB, the advanced system features include a 66-byte TX/RX FIFO, configurable automatic packet handler, listen mode, temperature sensor and configurable DIOs which greatly enhance system flexibility whilst at the same time significantly reducing MCU requirements.

The transceiver complies with both ETSI and FCC regulatory requirements

5.2 Simplified Block Diagram

Figure 5-1 shows a simplified block diagram of the MKW01Z128 transceiver.

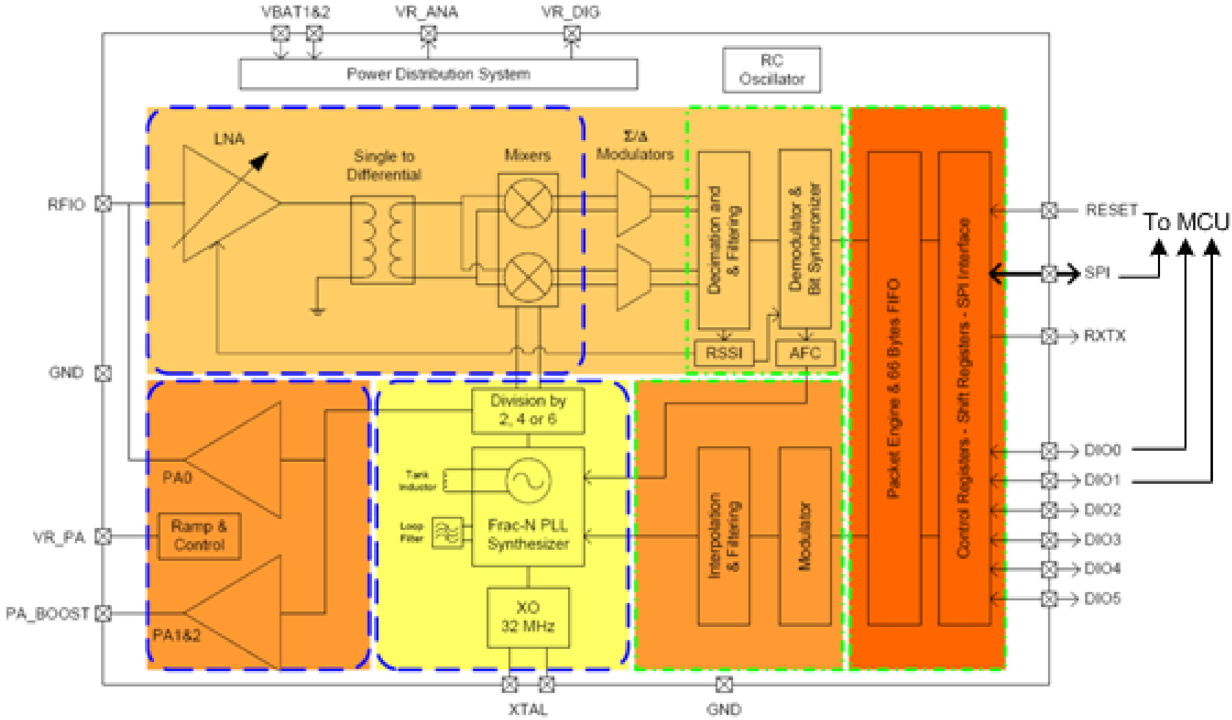


Figure 5-1. MKW01Z128 Transceiver Block Diagram

5.3 Transceiver Power Supply

The MKW01Z128 has separate power pins for both the MCU and the transceiver. The transceiver employs on-chip regulation and management that provides stable operating characteristics over the full temperature and voltage range of operation. This includes the full output power of +17dBm which is maintained from 1.8 to 3.6 V.

The transceiver voltage source is supplied via pins VBAT1 and VBAT2. See Section 4.2, “Power connections” for power supply connection and bypassing details. Freescale recommends that the MCU and transceiver be powered together from the same supply.

5.4 Low Battery Detector

A low battery detector is also included allowing the generation of an interrupt signal in response to passing a programmable threshold adjustable through the register *RegLowBat*. The interrupt signal can be mapped to any of the DIO pins, through the programming of *RegDioMapping*.

5.5 Frequency Synthesis

The LO generation on the MKW01Z128 transceiver is based on a state-of-the-art fractional-N PLL. The PLL is fully integrated with automatic calibration.

5.5.1 Reference Oscillator

The crystal oscillator is the main timing reference of the MKW01Z128. It is used as a reference for the transceiver frequency synthesizer and as a clock for the digital processing. In turn, the transceiver ClkOut signal can be used to provide an external reference clock for the MCU (see Section 4.4, “System Clock Sources and Configurations”).

The XO startup time, TS_OSC, depends on the actual XTAL being connected on pins XTA and XTB. When using the built-in sequencer, the device optimizes the startup time and automatically triggers the PLL when the XO signal is stable. To manually control the startup time, the user should either wait for TS_OSC max, or monitor the signal CLKOUT which will only be made available on the output buffer when a stable XO oscillation is achieved.

An external clock can be used to replace the crystal oscillator, for instance a tight tolerance TCXO. To do so, Bit 4 at transceiver Address 0x59 should be set to 1, and the external clock has to be provided on XTA (Pin 4). XTB (pin 5) should be left open. The peak-peak amplitude of the input signal must never exceed 1.8 V. Please consult your TCXO supplier for an appropriate value of decoupling capacitor, C_D.

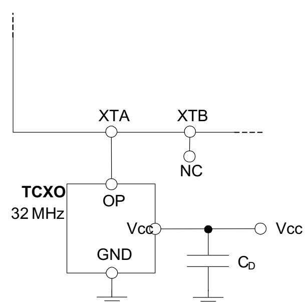


Figure 5-2. TCXO Connection

5.5.2 CLKOUT Output

The reference frequency, or a fraction of it, can be provided on DIO5 (pin 54) by modifying bits *ClkOut* in *RegDioMapping2*. Two typical applications of the CLKOUT output include:

- Provide a clock output for the MCU (see Section 4.4, “System Clock Sources and Configurations”) - saves the cost of an additional crystal. CLKOUT can be made available in any operation mode except Sleep mode and is automatically enabled at power-on reset.
- Provides an oscillator reference output - allows simple software trimming of the initial crystal tolerance.

If the transceiver is put into low power mode, ClkOut may be disabled for lowest power.

5.5.3 PLL Architecture

The frequency synthesizer generating the LO frequency for both the receiver and the transmitter is a fractional-N sigma-delta PLL. The PLL incorporates a third order loop capable of fast auto-calibration, and it has a fast switching-time. The VCO and the loop filter are both fully integrated, removing the need for an external tight-tolerance, high-Q inductor in the VCO tank circuit.

5.5.3.1 VCO

The VCO runs at 2, 4 or 6 times the RF frequency (respectively in the 915, 434 and 315 MHz bands) to reduce any LO leakage in receiver mode, to improve the quadrature precision of the receiver, and to reduce the pulling effects on the VCO during transmission.

The VCO calibration is fully automated. A coarse adjustment is carried out at power-on reset, and fine tuning is performed each time the transceiver PLL is activated. Automatic calibration times are fully transparent to the end-user, as this processing time is included in the *TS_TE* and *TS_RE* specifications.

5.5.3.2 PLL Bandwidth

The bandwidth of the Fractional-N PLL is wide enough to allow for:

- High speed FSK modulation, up to 600 kb/s, inside the PLL bandwidth
- Very fast PLL lock times - enabling both short startup and fast hop times required for frequency agile applications

5.5.3.3 Carrier Frequency and Resolution

The transceiver PLL embeds a 19-bit sigma-delta modulator and its frequency resolution, constant over the whole frequency range, and is given by:

$$F_{STEP} = \frac{F_{XOSC}}{2^{19}}$$

The carrier frequency is programmed through *RegFrf*, split across Addresses 0x07 to 0x09:

$$F_{RF} = F_{STEP} \times Frf(23,0)$$

NOTE

The Frf setting is split across 3 bytes. A change in the center frequency will only be taken into account when the least significant byte *FrfLsb* in *RegFrfLsb* is written. This allows for more complex modulation schemes such as m-ary FSK, where frequency modulation is achieved by changing the programmed RF frequency.

5.5.4 Lock Time

PLL lock time *TS_FS* is a function of a number of technical factors, such as synthesized frequency, frequency step, etc. When using the built-in sequencer, the transceiver optimizes the startup time and automatically starts the receiver or the transmitter when the PLL has locked. To manually control the startup time, the user should either wait for *TS_FS* max given in the specification, or monitor the signal PLL lock detect indicator, which is set when the PLL has is within its locking range.

When performing an AFC, which usually corrects very small frequency errors, the PLL response time is approximately:

$$T_{PLLAFC} = \frac{5}{PLLBW}$$

PLLBW default = Varies by region; see Table 7-11 for settings.

In a frequency hopping scheme, the timings *TS_HOP* given in the table of specifications give an order of magnitude for the expected lock times.

5.5.5 Lock Detect Indicator

A lock indication signal can be made available on some of the DIO pins, and is toggled high when the PLL reaches its locking range.

5.6 Transmitter Description

The transmitter of MKW01Z128 transceiver is comprised of the frequency synthesizer, modulator and power amplifier blocks.

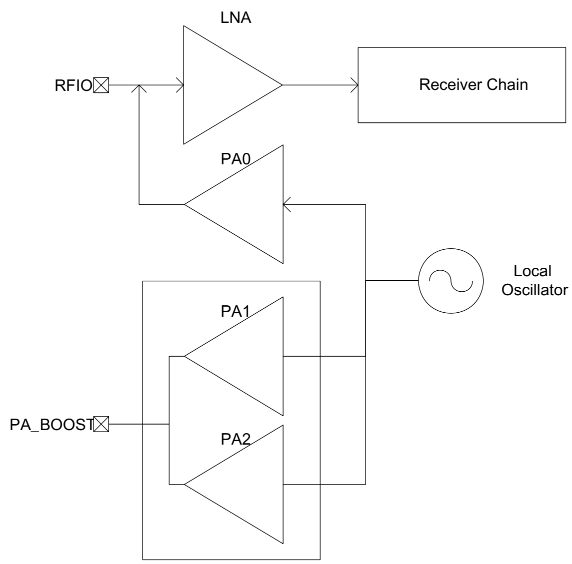


Figure 5-3. Transmitter Block Diagram

5.6.1 Bit Rate Setting

When using the transceiver in Continuous mode, the data stream to be transmitted can be input directly to the modulator via pin DIO2/DATA in an asynchronous manner, unless Gaussian filtering is used, in which case the DCLK signal on pin 10 (DIO1/DCLK) is used to synchronize the data stream.

In Packet mode or in Continuous mode with Gaussian filtering enabled, the Bit Rate (BR) is controlled by bits *BitRate* in *RegBitrate*:

$$BR = \frac{F_{XOSC}}{BitRate}$$

Among others, the following Bit Rates are accessible:

Table 5-1. Bit Rate Examples (for 32 MHz reference)

Type	BitRate (15:8)	BitRate (7:0)	(G)FSK (G)MSK	OOK	Actual BR (b/s)
Classical transceiver baud rates (multiples of 1.2 kbps)	0x68	0x2B	1.2 kbps	1.2 kbps	1199.985
	0x34	0x15	2.4 kbps	2.4 kbps	2400.060
	0x1A	0x0B	4.8 kbps	4.8 kbps	4799.760
	0x0D	0x05	9.6 kbps	9.6 kbps	9600.960
	0x06	0x83	19.2 kbps	19.2 kbps	19196.16
	0x03	0x41	38.4 kbps		38415.37
	0x01	0xA1	76.8 kbps		76738.61
	0x00	0xD0	153.6 kbps		153846.15
Classical transceiver baud rates (multiples of 0.9 kbps)	0x02	0x2C	57.6 kbps		57553.96
	0x01	0x16	115.2 kbps		115107.9
Round bit rates (multiples of 12.5, 25 and 50 kbps)	0x0A	0x00	12.5 kbps	12.5 kbps	12500.00
	0x05	0x00	25 kbps	25 kbps	25000.00
	0x02	0x80	50 kbps		50000.00
	0x01	0x40	100 kbps		100000.0
	0x00	0xD5	150 kbps		150234.74
	0x00	0xA0	200 kbps		200000.0
	0x00	0x80	250 kbps		250000.0
	0x00	0x6B	300 kbps		299065.4
	0x00	0x40	500 kbps		
	0x00	0x35	600 kbps		
Watch Xtal frequency	0x03	0xD1	32.768 kbps	32.768 kbps	32753.33

5.6.2 FSK Modulation

FSK modulation is performed inside the PLL bandwidth, by changing the fractional divider ratio in the feedback loop of the PLL. The large resolution of the sigma-delta modulator, allows for very narrow frequency deviation. The frequency deviation F_{DEV} is given by:

$$F_{DEV} = F_{STEP} \times Fdev(13,0)$$

To ensure a proper modulation, the following limit applies:

$$F_{DEV} + \frac{BR}{2} \le 500\text{kHz}$$

NOTE

No constraint applies to the modulation index of the transmitter, but the frequency deviation F_{DEV} must exceed 600 Hz.

5.6.3 OOK Modulation

OOK modulation is applied by switching on and off the Power Amplifier. Digital control and smoothing are available to improve the transient power response of the OOK transmitter.

5.6.4 Modulation Shaping

Modulation shaping can be applied in both OOK and FSK modulation modes, to improve the narrowband response of the transmitter. Both shaping features are controlled with *PaRamp* bits in *RegPaRamp*.

- In FSK mode, a Gaussian filter with BT = 0.3, 0.5 or 1 is used to filter the modulation stream, at the input of the sigma-delta modulator. If the Gaussian filter is enabled when the MKW01Z128 is in Continuous mode, DCLK signal on pin 10 (DIO1/DCLK) will trigger an interrupt on the MCU each time a new bit has to be transmitted.
- When OOK modulation is used, the PA bias voltages are ramped up and down smoothly when the PA is turned on and off, to reduce spectral splatter.

NOTE

The transmitter must be restarted if the ModulationShaping setting is changed, in order to recalibrate the built-in filter.

5.6.5 Power Amplifiers

Three power amplifier blocks are embedded in the transmitter. The first one (PA0) can generate up to +13 dBm into a 50 Ohm load. PA0 shares a common front-end pin RFIO with the receiver LNA.

PA1 and PA2 are both connected to pin PA_BOOST, allowing for two distinct power ranges:

- Low power mode - where power out is -18 dBm < Pout < 13 dBm, with PA1 enabled
- Higher power mode - when PA1 and PA2 are combined, providing up to +17 dBm to a matched load.

NOTE

When PA1 and PA2 are combined to deliver +17 dBm to the antenna, a specific impedance matching / harmonic filtering design is required to ensure impedance transformation and regulatory compliance.

All PA settings are controlled by *RegPaLevel*, and the truth table of settings is given in Table 5-2.

Table 5-2. Power Amplifier Mode Selection Truth Table

Pa0On	Pa1On	Pa2On	Mode	Power Range	Pout Formula
1	0	0	PA0 output on pin RFIO	-18 to +13 dBm	-18 dBm + *OutputPower*
0	1	0	PA1 enabled on pin PA_BOOST	-18 to +13 dBm	-18 dBm + *OutputPower*
0	1	1	PA1 and PA2 combined on pin PA_BOOST	+2 to +17 dBm	-14 dBm + *OutputPower*[1]
Other combinations			Reserved		

[1] Output power should be used only for the 16 steps from 0x10 to 0x1F. The range of 0x00 to 0x0F puts PA1 on the lower half of its 32 step range but PA2 has only 16 steps in the upper half of the power curve regardless of the state of bit 4.

NOTE

- To ensure correct operation at the highest power levels, adjust the Over Current Protection Limit accordingly in RegOcp.
- If PA_BOOST pin is not used, the pin can be left floating.

5.6.6 Over Current Protection

An over-current protection block is built-in the chip that helps prevent surge currents when the transmitter is used at its highest power levels, thus protecting the battery that may power the application. The current clamping value is controlled by *OcpTrim* bits in *RegOcp*, and is calculated with the following formula:

$$\mathrm{Imax} = 45 + 5 \times \mathrm{OcpTrim(mA)}$$

NOTE

Imax sets the maximum current drawn by the final PA stage, and does not account for the PA drivers and frequency synthesizer. Global current drain on Vbatt will be higher

5.7 Receiver Description

The MKW01Z128 transceiver features a digital receiver with the analog to digital conversion process being performed directly following the LNA-Mixers block. The zero-IF receiver is able to handle (G)FSK and (G)MSK modulation. ASK and OOK modulation is, however, demodulated by a low-IF architecture. All the filtering, demodulation, gain control, synchronization and packet handling is performed digitally, and this allows a very wide range of bit rates and frequency deviations to be selected. The receiver is also capable of automatic gain calibration in order to improve precision of RSSI measurements.

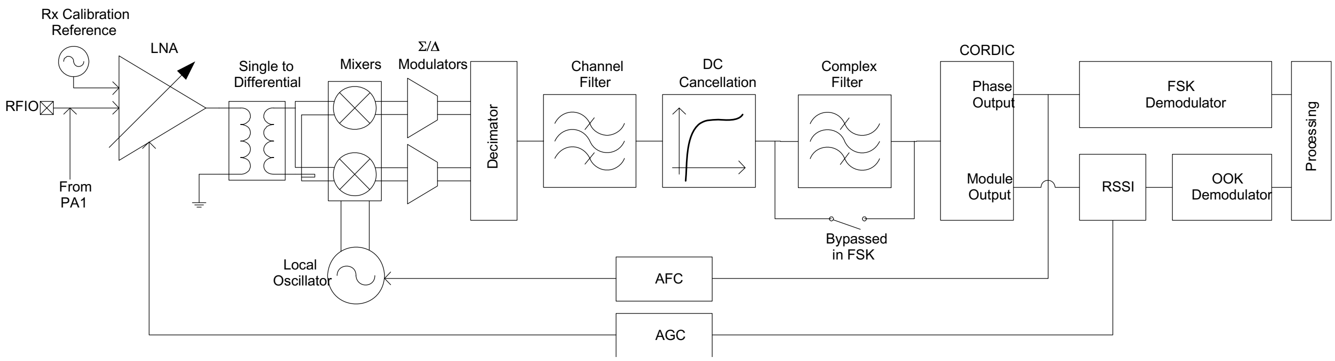


Figure 5-4. Receiver Block Diagram

The following sections give a brief description of each of the receiver blocks.

5.7.1 LNA - Single to Differential Buffer

The LNA uses a common-gate topology, which allows for a flat characteristic over the whole frequency range. It is designed to have an input impedance of 50 Ohms or 200 Ohms (as selected with bit *LnaZin* in *RegLna)*, and the parasitic capacitance at the LNA input port is cancelled with an external RF choke. A single to differential buffer is implemented to improve the second order linearity of the receiver.

NOTE

Due to circuit specifics and matching topology, optimum performance in any given design may be achieved with LNA Zin set to either 50 Ω or 200 Ω. Both settings should be tested for sensitivity in the actual system in development.

The LNA gain, including the single-to-differential buffer, is programmable over a 48 dB dynamic range, and control is either manual or automatic with the embedded AGC function.

NOTE

In the specific case where the LNA gain is manually set by the user, the receiver will not be able to properly handle FSK signals with a modulation index smaller than 2 at an input power greater than the 1dB compression point ($P1_{dB}$), described in Table 5-4.

Table 5-3. LNA Gain Settings

LnaGainSelect	LNA Gain	Gain Setting
000	Any of the below, set by the AGC loop	-
001	Max gain	G1
010	Max gain - 6 dB	G2
011	Max gain - 12 dB	G3
100	Max gain - 24 dB	G4

Table 5-3. LNA Gain Settings

101	Max gain - 36 dB	G5
110	Max gain - 48 dB	G6
111	Reserved	-

5.7.2 Automatic Gain Control

By default (*LnaGainSelect = 000*), the LNA gain is controlled by a digital AGC loop in order to obtain the optimal sensitivity/linearity trade-off.

Regardless of the data transfer mode (Packet or Continuous), the following series of events takes place when the receiver is enabled:

- The receiver stays in WAIT mode, until *RssiValue* exceeds *RssiThreshold* for two consecutive samples. Its power consumption is the receiver power consumption.
- When this condition is satisfied, the receiver automatically selects the most suitable LNA gain, optimizing the sensitivity/linearity trade-off.
- The programmed LNA gain, read-accessible with *LnaCurrentGain* in *RegLna*, is carried on for the whole duration of the packet, until one of the following conditions is fulfilled:
  - Packet mode: if *AutoRxRestartOn = 0,* the LNA gain will remain the same for the reception of the following packet. If *AutoRxRestartOn = 1,* after the controller has emptied the FIFO the receiver will re-enter the WAIT mode described above, after a delay of *InterPacketRxDelay,* allowing for the distant transmitter to ramp down, hence avoiding a false RSSI detection. In both cases (*AutoRxRestartOn=0* or *AutoRxRestartOn=1*), the receiver can also re-enter the WAIT mode by setting RestartRx bit to 1. The user can decide to do so, to manually launch a new AGC procedure.
  - Continuous mode: upon reception of valid data, the user can decide to either leave the receiver enabled with the same LNA gain, or to restart the procedure, by setting *RestartRx* bit to 1, resuming the WAIT mode of the receiver, described above.

NOTE

- The AGC procedure must be performed while receiving preamble in FSK mode.
- In OOK mode, the AGC will give better results if performed while receiving a constant “1” sequence

The following figure illustrates the AGC behavior:

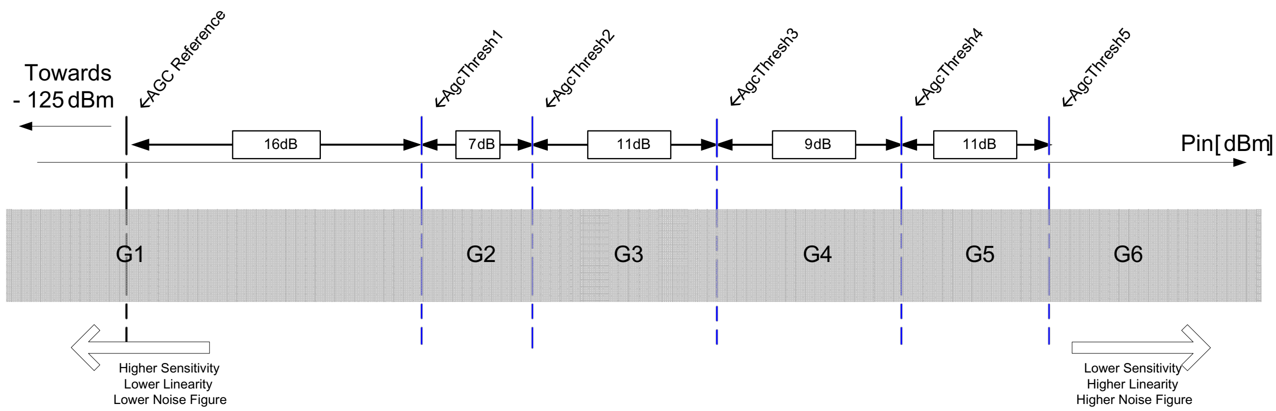


Figure 5-5. AGC Thresholds Settings

The following table summarizes the typical performance of the complete receiver:

Table 5-4. Receiver Performance Summary

Input Power Pin	Gain Setting	Receiver Performance (typ)			
		P_{-1dB} [dBm]	NF [dB]	IIP3 [dBm]	IIP2 [dBm]
Pin < AgcThresh1	G1	-37	7	-18	+35
AgcThresh1 < Pin < AgcThresh2	G2	-31	13	-15	+40
AgcThresh2 < Pin < AgcThresh3	G3	-26	18	-8	+48
AgcThresh3 < Pin < AgcThresh4	G4	-14	27	-1	+62
AgcThresh4 < Pin < AgcThresh5	G5	>-6	36	+13	+68
AgcThresh5 < Pin	G6	>0	44	+20	+75

5.7.2.1 RssiThreshold Setting

For correct operation of the AGC, *RssiThreshold* in *RegRssiThresh* must be set to the sensitivity of the receiver. The receiver will remain in WAIT mode until *RssiThreshold* is exceeded.

NOTE

When AFC is enabled and performed automatically at the receiver startup, the channel filter is used by the receiver during the AFC and the AGC is RxBwAfc instead of the standard RxBw setting. This may impact the sensitivity of the receiver, and the setting of RssiThreshold accordingly.

5.7.2.2 AGC Reference

The AGC reference level is automatically computed in the transceiver according to:

AGC Reference [dBm] = -174 + NF + DemodSnr +10*log(2*RxBw) + FadingMargin [dBm]

With:

- *NF* = 7dB : LNA's Noise Figure at maximum gain
- *DemodSnr* = 8 dB : SNR needed by the demodulator
- *RxBw* : Single sideband channel filter bandwidth
- FadingMargin = 5 dB : Fading margin

5.7.3 Continuous-Time DAGC

In addition to the automatic gain control described in Section 5.7.2, "Automatic Gain Control", the transceiver is capable of continuously adjusting its gain in the digital domain, after the analog to digital conversion has occurred. This feature, named DAGC, is fully transparent to the end user. The digital gain adjustment is repeated every 2 bits, and has the following benefits:

- Fully transparent to the end user
- Improves the fading margin of the receiver during the reception of a packet, even if the gain of the LNA is frozen
- Improves the receiver robustness in fast fading signal conditions, by quickly adjusting the receiver gain (every 2 bits)
- Works in Continuous, Packet, and unlimited length Packet modes

The DAGC is enabled by setting *RegTestDagc* to 0x10 for low modulation index systems (i.e. when *AfcLowBetaOn=1*) and 0x30 for other systems. It is recommended to always enable the DAGC.

5.7.4 Quadrature Mixer - ADCs - Decimators

The mixer is inserted between output of the RF buffer stage and the input of the analog to digital converter (ADC) of the receiver section. This block is designed to translate the spectrum of the input RF signal to baseband, and offer both high IIP2 and IIP3 responses.

In the lower bands of operation (290 to 510 MHz), the multi-phase mixing architecture with weighted phases improves the rejection of the LO harmonics in receiver mode, hence increasing the receiver immunity to out-of-band interferers.

The I and Q digitalization is made by two 5^{th} order continuous-time Sigma-Delta Analog to Digital Converters (ADC). Their gain is not constant over temperature, but the whole receiver is calibrated before reception, so that this inaccuracy has no impact on the RSSI precision. The ADC output is one bit per channel. It needs to be decimated and filtered afterwards. This ADC can also be used for temperature measurement; please refer to Section 5.7.16, "Temperature Sensor" for more details.

The decimators decrease the sample rate of the incoming signal in order to optimize the area and power consumption of the following receiver blocks.

5.7.5 Channel Filter

The role of the channel filter is to filter out the noise and interferers outside of the channel. Channel filtering on the transceiver is implemented with a 16-tap Finite Impulse Response (FIR) filter, providing an outstanding Adjacent Channel Rejection performance, even for narrowband applications.

NOTE

To respect oversampling rules in the decimation chain of the receiver, the Bit Rate cannot be set to a value higher than 2 times the single-side receiver bandwidth (BitRate < 2 x RxBw)

The single-side channel filter bandwidth RxBw is controlled by the parameters RxBwMant and RxBwExp in RegRxBw:

- When FSK modulation is enabled:

$$RxBw = \frac{FXOSC}{RxBwMant \times 2^{RxBwExp + 2}}$$

- When OOK modulation is enabled:

$$RxBw = \frac{FXOSC}{RxBwMant \times 2^{RxBwExp + 3}}$$

NOTE

The following channel filter bandwidths are accessible (oscillator is mandated at 32 MHz):

Table 5-5. Available RxBw Settings

RxBwMant (binary/value)	RxBwExp (decimal)	RxBw (kHz)	
		FSK ModulationType=00	OOK ModulationType=01
10b / 24	7	2.6	1.3
01b / 20	7	3.1	1.6
00b / 16	7	3.9	2.0
10b / 24	6	5.2	2.6
01b / 20	6	6.3	3.1
00b / 16	6	7.8	3.9
10b / 24	5	10.4	5.2
01b / 20	5	12.5	6.3
00b / 16	5	15.6	7.8
10b / 24	4	20.8	10.4
01b / 20	4	25.0	12.5
00b / 16	4	31.3	15.6
10b / 24	3	41.7	20.8

Table 5-5. Available RxBw Settings

01b / 20	3	50.0	25.0
00b / 16	3	62.5	31.3
10b / 24	2	83.3	41.7
01b / 20	2	100.0	50.0
00b / 16	2	125.0	62.5
10b / 24	1	166.7	83.3
01b / 20	1	200.0	100.0
00b / 16	1	250.0	125.0
10b / 24	0	333.3	166.7
01b / 20	0	400.0	200.0
00b / 16	0	500.0	250.0

5.7.6 DC Cancellation

DC cancellation is required in zero-IF architecture transceivers to remove any DC offset generated through self-reception. It is built into the device and its adjustable cutoff frequency *fc* is controlled in *RegRxBw*:

$$fc = \frac{4 \times RxBw}{2\pi \times 2^{DccFreq + 2}}$$

The default value of *DccFreq* cutoff frequency is typically 4% of the RxBw (channel filter BW). The cutoff frequency of the DCC can however be increased to slightly improve the sensitivity under wider modulation conditions. It is advised to adjust the DCC setting while monitoring the receiver sensitivity.

5.7.7 Complex Filter - OOK

In OOK mode the receiver is modified to a low-IF architecture. The IF frequency is automatically set to half the single side bandwidth of the channel filter (F_{IF} = 0.5 x *RxBw*). The Local Oscillator is automatically offset by the IF in the OOK receiver. A complex filter is implemented on the chip to attenuate the resulting image frequency by typically 30 dB.

NOTE

This filter is automatically bypassed when receiving FSK signals (ModulationType = 00 in RegDataModul).

5.7.8 RSSI

The RSSI block evaluates the amount of energy available within the receiver channel bandwidth. Its resolution is 0.5 dB, and it has a wide dynamic range to accommodate both small and large signal levels that may be present. Its acquisition time is very short, taking only 2 bit periods. The RSSI sampling must occur during the reception of preamble in FSK, and constant “1” reception in OOK.

NOTE

- The receiver is capable of automatic gain calibration in order to improve the precision of its RSSI measurements. This function injects a known RF signal at the LNA input, and calibrates the receiver gain accordingly. This calibration is automatically performed during the PLL start-up, making it a transparent process to the end-user
- *RssiValue* can only be read when it exceeds *RssiThreshold*

5.7.9 Cordic

The Cordic task is to extract the phase and the amplitude of the modulation vector (I+j.Q). This information, still in the digital domain is used as follows:

- Phase output - used by the FSK demodulator and the AFC blocks.
- Amplitude output - used by the RSSI block, for FSK demodulation, AGC and automatic gain calibration purposes.

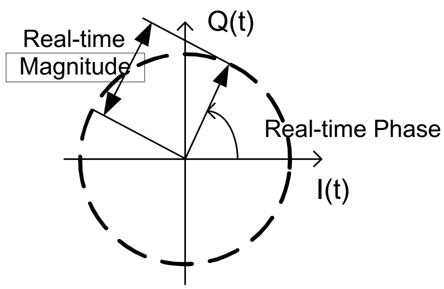


Figure 5-6. Cordic Extraction

5.7.10 FSK Demodulator

The FSK demodulator of the receiver is designed to demodulate FSK, GFSK, MSK and GMSK modulated signals. It is most efficient when the modulation index of the signal is greater than 0.5 and below 10:

$$0.5 \le \beta = \frac{2 \times F_{DEV}}{BR} \le 10$$

The output of the FSK demodulator can be fed to the Bit Synchronizer (described in Section 5.7.12), to provide the companion processor with a synchronous data stream in Continuous mode.

5.7.11 OOK Demodulator

The OOK demodulator performs a comparison of the RSSI output and a threshold value. Three different threshold modes are available, configured through bits *OokThreshType* in *RegOokPeak*.

The recommended mode of operation is the "Peak" threshold mode, illustrated in Figure 5-7:

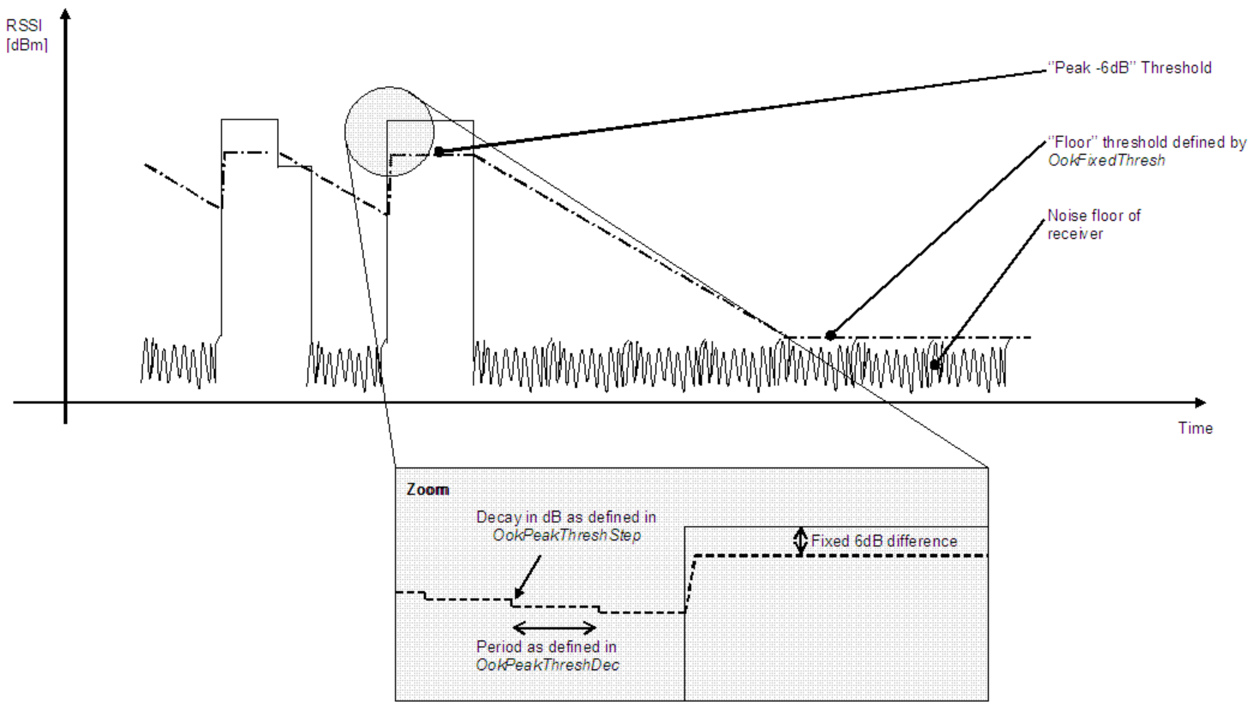


Figure 5-7. OOK Peak Demodulator Description

In peak threshold mode the comparison threshold level is the peak value of the RSSI, reduced by 6dB. In the absence of an input signal, or during the reception of a logical "0", the acquired peak value is decremented by one *OokPeakThreshStep* every *OokPeakThreshDec* period.

When the RSSI output is null for a long time (for instance after a long string of "0" received, or if no transmitter is present), the peak threshold level will continue falling until it reaches the "Floor Threshold", programmed in *OokFixedThresh*.

The default settings of the OOK demodulator lead to the performance stated in the electrical specification. However, in applications in which sudden signal drops are awaited during a reception, the three parameters should be optimized accordingly.

5.7.11.1 Optimizing the Floor Threshold

OokFixedThresh determines the sensitivity of the OOK receiver, as it sets the comparison threshold for weak input signals (i.e., those close to the noise floor). Significant sensitivity improvements can be generated if configured correctly.

Note that the noise floor of the receiver at the demodulator input depends on:

- The noise figure of the receiver.
- The gain of the receive chain from antenna to base band.
- The matching - including SAW filter if any.
- The bandwidth of the channel filters.

It is therefore important to note that the setting of *OokFixedThresh* will be application dependant. The following procedure is recommended to optimize *OokFixedThresh*.

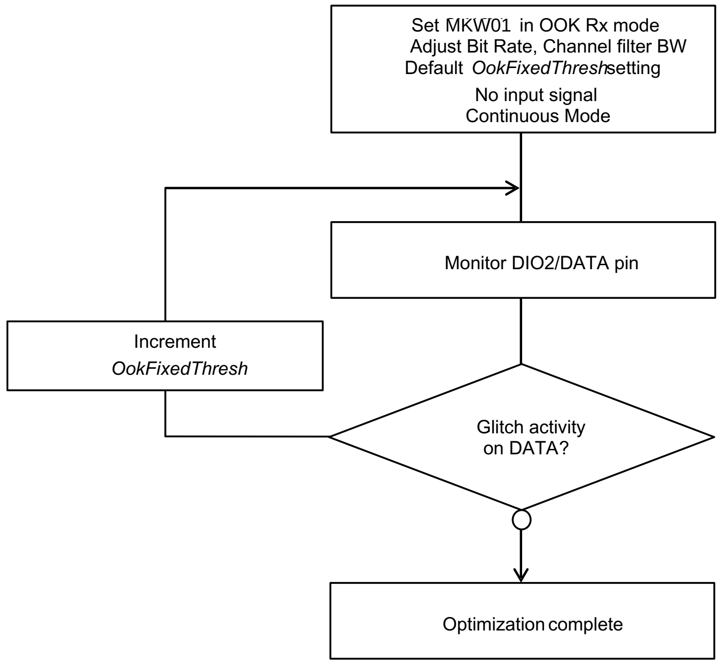


Figure 5-8. Floor Threshold Optimization

The new floor threshold value found during this test should be used for OOK reception with those receiver settings.

5.7.11.2 Optimizing OOK Demodulator for Fast Fading Signals

A sudden drop in signal strength can cause the bit error rate to increase. For applications where the expected signal drop can be estimated, the following OOK demodulator parameters *OokPeakThreshStep* and *OokPeakThreshDec* can be optimized as described below for a given number of threshold decrements per bit. Refer to *RegOokPeak* to access those settings.

5.7.11.3 Alternative OOK Demodulator Threshold Modes

In addition to the Peak OOK threshold mode, the user can alternatively select two other types of threshold detectors:

- Fixed Threshold: - The value is selected through *OokFixedThresh*
- Average Threshold - Data supplied by the RSSI block is averaged, and this operation mode should only be used with DC-free encoded data.

5.7.12 Bit Synchronizer

The Bit Synchronizer is a block that provides a clean and synchronized digital output, free of glitches. Its output is made available on pin DIO1/DCLK in Continuous mode and can be disabled through register settings. However, for optimum receiver performance its use when running Continuous mode is strongly advised.

The Bit Synchronizer is automatically activated in Packet mode. Its bit rate is controlled by *BitRateMsb* and *BitRateLsb* in *RegBitrate.*

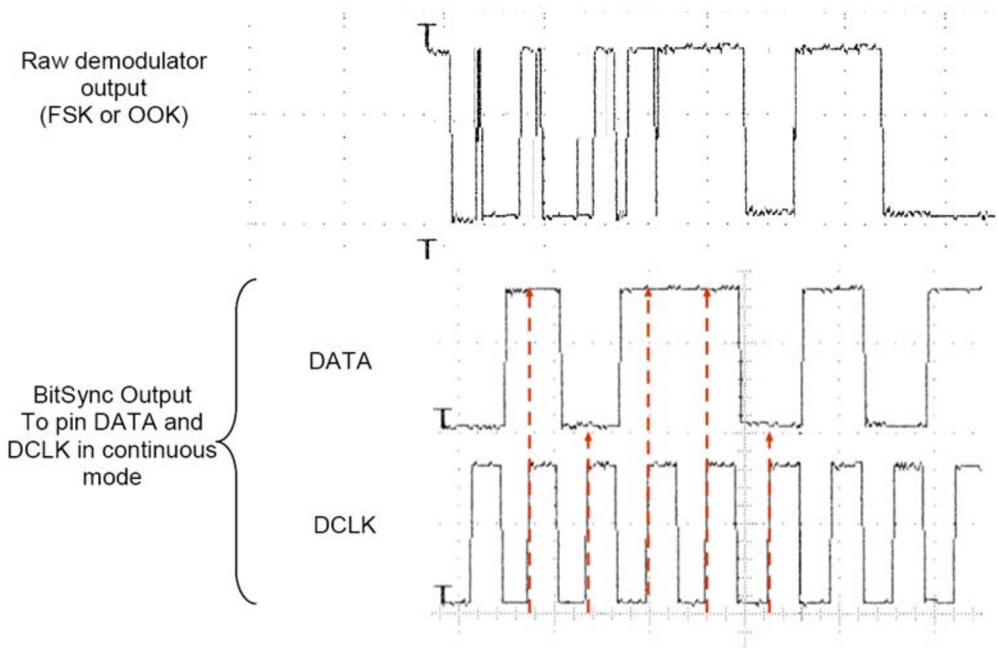


Figure 5-9. Bit Synchronizer Description

To ensure correct operation of the Bit Synchronizer, the following conditions have to be satisfied:

- A preamble (0x55 or 0xAA) of 12 bits is required for synchronization (from the *RxReady* interrupt)
- The subsequent payload bit stream must have at least one transition form '0' to '1' or '1' to '0 every 16 bits during data transmission
- The bit rate matching between the transmitter and the receiver must be better than 6.5 %.

NOTE

- If the Bit Rates of transmitter and receiver are known to be the same, the receiver will be able to receive an infinite unbalanced sequence (all “0s” or all ”1s”) with no restriction.
- If there is a difference in Bit Rate between TX and RX, the amount of adjacent bits at the same level that the BitSync can withstand can be estimated as follows:

$$NumberOfBits = \frac{1}{2} * \frac{BR}{\Delta BR}$$

- This implies approximately 6 consecutive unbalanced bytes when the Bit Rate precision is 1%, which is easily achievable (crystal tolerance is in the range of 50 to 100 ppm).

5.7.13 Frequency Error Indicator (FEI)

This function provides information about the frequency error of the local oscillator (LO) compared with the carrier frequency of a modulated signal at the input of the receiver. When the FEI block is launched, the frequency error is measured and the signed result is loaded in *FeiValue* in *RegFei*, in 2's complement format. The time required for an FEI evaluation is 4 times the bit period.

To ensure a proper behavior of the FEI:

- The operation must be done during the reception of preamble
- The sum of the frequency offset and the 20 dB signal bandwidth must be lower than the base band filter bandwidth

The 20 dB bandwidth of the signal can be evaluated as follows (double-side bandwidth):

$$BW_{20dB} = 2 \times \left(F_{DEV} + \frac{BR}{2}\right)$$

The frequency error, in Hz, can be calculated with the following formula:

$$FEI = F_{STEP} \times FeiValue$$

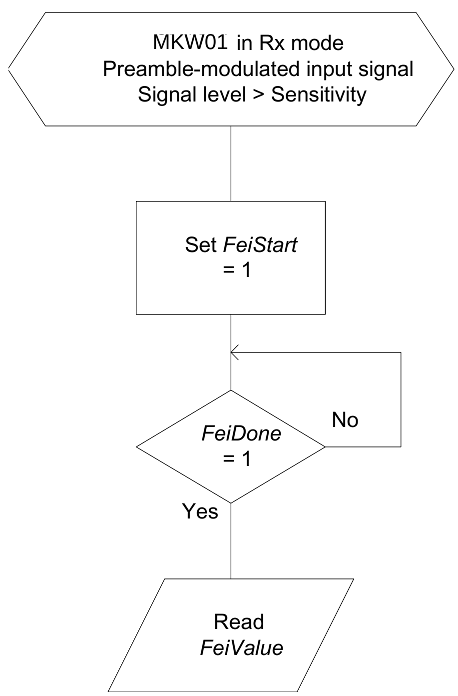


Figure 5-10. FEI Process

5.7.14 Automatic Frequency Correction (AFC)

The AFC is based on the FEI block, and therefore the same input signal and receiver setting conditions apply. When the AFC procedure is done, *AfcValue* is directly subtracted to the register that defines the frequency of operation of the chip, F_{RF}. The AFC can be launched:

- Each time the receiver is enabled, if *AfcAutoOn* = 1
- Upon user request, by setting bit *AfcStart* in *RegAfcFei*, if *AfcAutoOn* = 0

When the AFC is automatically triggered (*AfcAutoOn* = 1), the user has the option to:

- Clear the former AFC correction value, if *AfcAutoClearOn* = 1
- Start the AFC evaluation from the previously corrected frequency. This may be useful in systems in which the LO keeps on drifting in the “same direction”. Aging compensation is a good example.

The receiver offers an alternate receiver bandwidth setting during the AFC phase, to accommodate large LO drifts. If the user considers that the received signal may be out of the receiver bandwidth, a higher channel filter bandwidth can be programmed in *RegAfcBw*, at the expense of the receiver noise floor, which will impact sensitivity.

5.7.15 Optimized Setup for Low Modulation Index Systems

The following apply for optimizing low modulation index systems:

- For wide band systems, where AFC is usually not required (XTAL inaccuracies do not typically impact the sensitivity), it is recommended to offset the LO frequency of the receiver to avoid desensitization. This can be simply done by modifying *Frf* in *RegFrfLsb*. A good generalization is to offset the receiver’s LO by 10% of the expected transmitter frequency deviation.
- For narrow band systems, it is recommended to perform AFC. The receiver has a dedicated AFC, enabled when *AfcLowBetaOn* in *RegAfcCtrl* is set to 1. A frequency offset, programmable through *LowBetaAfcOffset* in *RegTestAfc*, is added and is calculated as follows:

$$\text{Offset} = \mathit{LowBetaAfcOffset} \times 488\ \text{Hz}$$

The user should ensure that the programmed offset exceeds the DC canceller’s cutoff frequency, set through *DccFreqAfc* in *RegAfcBw*.

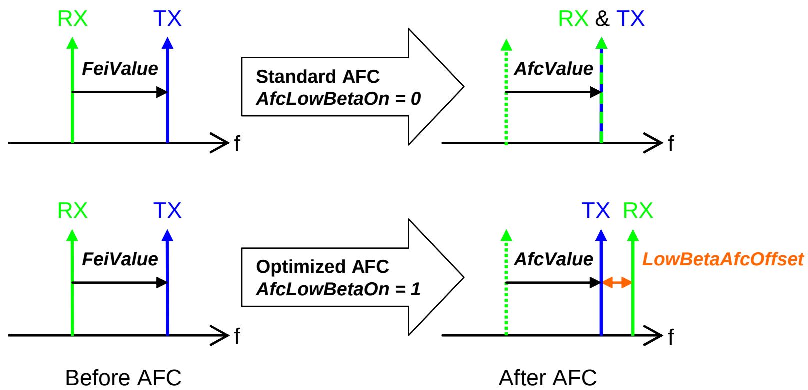


Figure 5-11. Optimized AFC (AfcLowBetaOn=1)

As shown on Figure 5-11, a standard AFC sequence uses the result of the FEI to correct the LO frequency and align both local oscillators. When the optimized AFC is enabled (*AfcLowBetaOn=1*), the receiver's LO is corrected by "*FeiValue* + *LowBetaAfcOffset*".

When the optimized AFC routine is enabled, the receiver startup time can be computed as follows:

TS_RE_AGC&AFC (optimized AFC) = *Tana* + *4.Tcf* + *4.Tdcc* + *3.Trssi* + *2.Tafc* + *2.Tpllafc*

5.7.16 Temperature Sensor

When temperature is measured, the receiver ADC is used to digitize the sensor response. Most receiver blocks are disabled, and temperature measurement can only be triggered in Standby or Frequency Synthesizer modes.

The response of the temperature sensor is -1°C / Lsb. A CMOS temperature sensor is not accurate by nature, therefore it should be calibrated at ambient temperature for precise temperature readings.

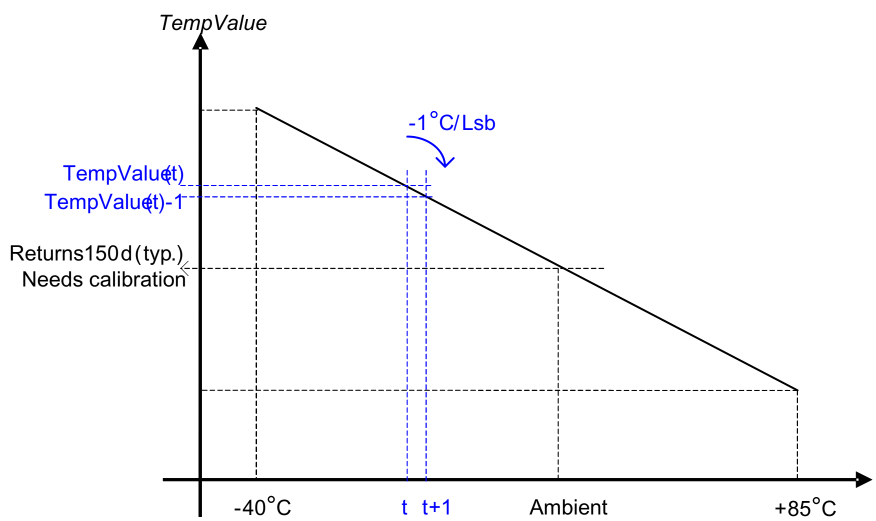


Figure 5-12. Temperature Sensor Response

It takes less than 100 microseconds for the transceiver to evaluate the temperature (from setting *TempMeasStart* to 1 to *TempMeasRunning* reset).

5.7.17 Timeout Function

The MKW01Z128 includes a Timeout function, which allows it to automatically shut-down the receiver after a receive sequence and therefore save energy.

- Timeout interrupt is generated *TimeoutRxStart x 8 x Tbit* after switching to RX mode if *RssiThreshold* flag does not raise within this time frame
- Timeout interrupt is generated *TimeoutRssiThresh x 8 x Tbit* after *RssiThreshold* flag has been raised.

This timeout interrupt can be used to warn the MCU to shut down the receiver and return to a lower power mode.

5.8 High Bit Rate Operations

High Bit rate operation is available in FSK mode. For operations in high bit rate, the frequency deviation should respect thefollowing equation: FDA + BRF/2 =<500kHz, where FDA is the Frequency Deviation and BR is the Bit Rate.

5.8.1 500 kbps Operation

For operation at 500 kbps, the following settings are recommended:

- FDA = 250kHz, where FDA is the Frequency Deviation (FSK operation with a Modulation index of 1)
- Crystal should be selected for a maximum of +/- 20ppm frequency stability.
- Carrier frequency of the receiver should be programmed with 50kHz offset from the programmed carrier frequency of transmitter. This offset takes into account the possible +/-20ppm drifts of Crystals. No AFC is needed.

5.8.2 600 kbps Operation

For operation at 600kbps, the following settings are recommended:

- FDA = 150kHz, where FDA is the Frequency Deviation (FSK operation with a Modulation index of 0.5)
- Crystal should be selected for a maximum of +/- 15ppm frequency stability.
- Carrier frequency of the receiver should be programmed with 40kHz offset from the programmed carrier frequency of transmitter. This offset takes into account the possible +/-15ppm drifts of Crystals. No AFC is needed.

NOTE

For both 500 and 600 kbps operations, RegTestPll must be set to 0x0C.

Chapter 6
Transceiver Operating Modes

This chapter describes the operating modes of the MKW01Z128 transceiver.

6.1 Basic Modes

The transceiver can be programmed to 5 different basic modes which are described in Table 6-1.

By default, when switching from a mode to another one, the sub-blocks are woken up according to a pre-defined and optimized sequence. Alternatively, these operating modes can be selected directly by disabling the automatic sequencer (*SequencerOff* in *RegOpMode = 1*).

Table 6-1. Basic Transceiver Modes

ListenOn in *RegOpMode*	Mode in *RegOpMode*	Selected mode	Enabled blocks
0	0 0 0	Sleep Mode	None
0	0 0 1	Stand-by Mode	Top regulator and crystal oscillator
0	0 1 0	FS Mode	Frequency synthesizer
0	0 1 1	Transmit Mode	Frequency synthesizer and transmitter
0	1 0 0	Receive Mode	Frequency synthesizer and receiver
1	x	Listen Mode	See Listen Mode, section Section 6.3, “Listen Mode”

6.2 Automatic Sequencer and Wake-Up Times

By default, when switching from one operating mode to another, the circuit takes care of the sequence of events in such a way that the transition timing is optimized. For example, when switching from Sleep mode to Transmit mode, the device goes first to Standby mode (XO started), then to frequency synthesizer mode, and finally, when the PLL has locked, to transmit mode. Entering transmit mode is also made according to a predefined sequence starting with the wake-up of the PA regulator before applying a ramp-up on the PA and generating the DCLK clock.

- The crystal oscillator wake-up time, TS_OSC, is directly related to the time for the crystal oscillator to reach its steady state. It depends notably on the crystal characteristics.
- The frequency synthesizer wake-up time, TS_FS, is directly related to the time needed by the PLL to reach its steady state. The signal PLL_LOCK, provided on an external pin, gives an indication of the lock status. It goes high when the PLL reaches its locking range.

Four specific cases can be highlighted:

- Transmitter Wake Up time from Sleep mode = *TS_OSC* + *TS_FS* + *TS_TR*

- Receiver Wake Up time from Sleep mode = *TS_OSC* + *TS_FS* + *TS_RE*
- Receiver Wake Up time from Sleep mode, AGC enabled = *TS_OSC* + *TS_FS* + *TS_RE_AGC*
- Receiver Wake Up time from Sleep mode, AGC and AFC enabled = *TS_OSC* + *TS_FS* + *TS_RE_AGC&AFC*

In applications where the target average power consumption, or the target startup time, do not require setting the transceiver in the lowest power modes (Sleep or Standby), the respective timings *TS_OSC* and *TS_FS* in the former equations can be omitted.

6.2.1 Transmitter Startup Time

The transmitter wake-up time, TS_TR, is given by the sequence controlled by the digital part. It is a pure digital delay which depends on the bit rate and the ramp-up time. In FSK mode, this time can be derived from the following equation.

$$TS\_TR = 5\mu s + 1.25 \times PaRamp + \frac{1}{2} \times Tbit$$

where *PaRamp* is the ramp-up time programmed in *RegPaRamp* and *Tbit* is the bit time.

In OOK mode, this equation can be simplified to the following:

$$TS\_TR = 5\mu s + \frac{1}{2} \times Tbit$$

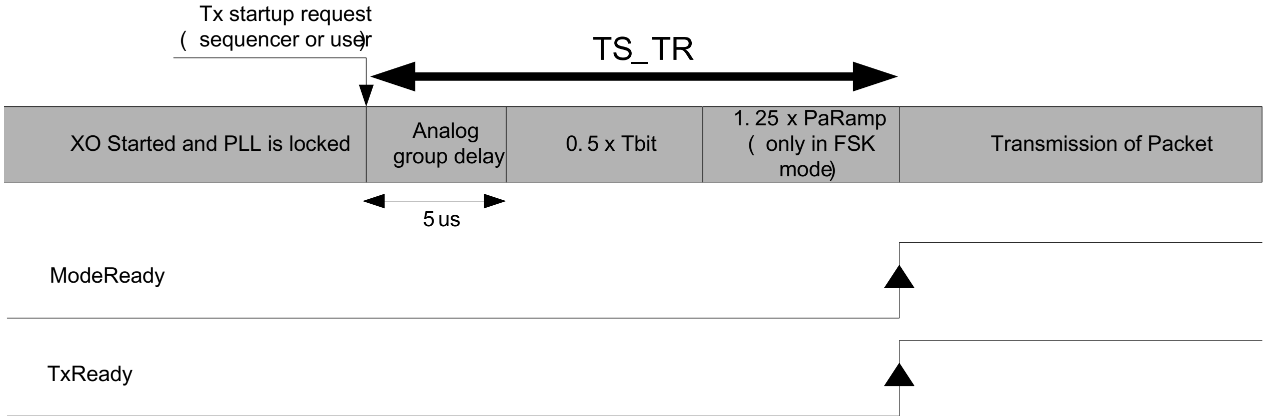


Figure 6-1. TX Startup, FSK and OOK

6.2.2 TX Start Procedure

As described in the former section, *ModeReady* and *TxReady* interrupts warn the MCU that the transmitter is ready to transmit data:

- In Continuous mode - the preamble bits preceding the payload can be applied on the DIO2/DATA pin immediately after any of these interrupts have fired. The DCLK signal, activated on pin DIO1/DCLK can also be used to start toggling the DATA pin, as described on Figure 6-2.

- In Packet mode - the transmitter will automatically modulate the RF signal with preamble bytes as soon as *TxReady* or *ModeReady* happen. The actual packet transmission (starting with the number of preambles specified in *PreambleSize*) will start when the *TxStartCondition* is fulfilled.

6.2.3 Receiver Startup Time

It is highly recommended to use the built-in sequencer of the transceiver to optimize the delays when setting the chip in receive mode. It guarantees the shortest startup times, hence the lowest possible energy usage, for battery operated systems.

The startup times of the receiver can be calculated from the following:

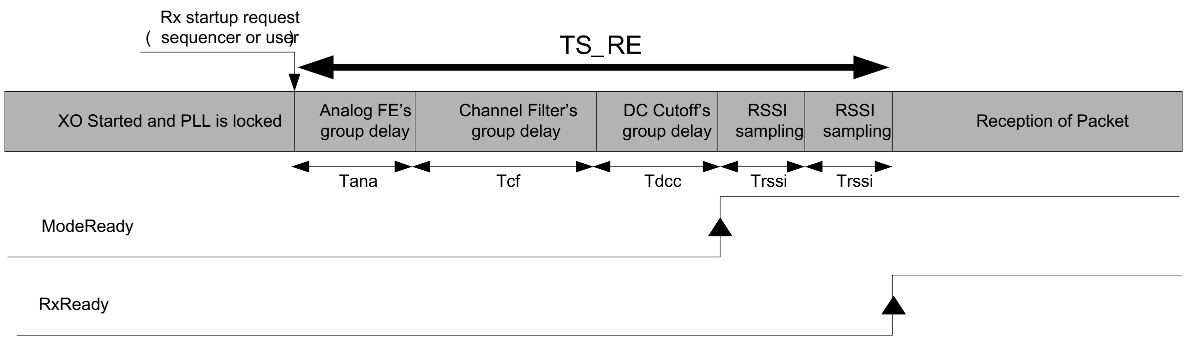


Figure 6-2. RX Startup - No AGC, no AFC

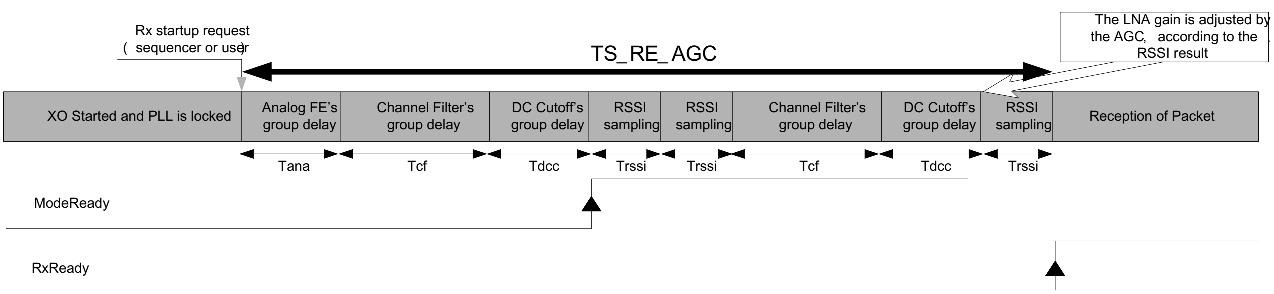


Figure 6-3. RX Startup - AGC, no AFC

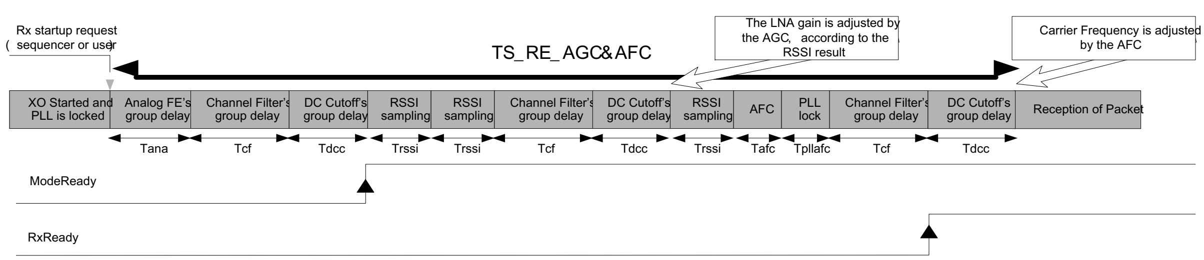


Figure 6-4. RX Startup - AGC and AFC

The different timings shown above are as follows:

- Group delay of the analog front end - *Tana = 20 us*
- Channel filter's group delay in FSK mode - *Tcf = 21 / (4.RxBw)*
- Channel filter's group delay in OOK mode - *Tcf = 34 / (4.RxBw)*

- DC Cutoff's group delay - *Tdcc = max(8 , 2^(round(log2(8.RxBw.Tbit)+1)) / (4.RxBw)*
- PLL lock time after AFC adjustment - *Tpllafc = 5 / PLLBW (PLLBW default = 300 kHz, see Table 7-11 for settings)*
- AFC sample time - *Tafc = 4 x Tbit* (also denoted TS_AFC in the general specification)
- RSSI sample time - *Trssi = 2 x int(4.RxBw.Tbit)/(4.RxBw)* (aka TS_RSSI)

NOTE

The above timings represent maximum settling times, and shorter settling times may be observed in real cases

6.2.4 RX Start Procedure

As described in the former sections, the *RxReady* interrupt warns the MCU that the receiver is ready.

- In Continuous mode with Bit Synchronizer, the receiver will start locking its Bit Synchronizer on a minimum or 12 bits of received preamble, before the reception of correct Data, or Sync Word (if enabled) can occur.
- In Continuous mode without Bit Synchronizer, valid data will be available on DIO2/DATA right after the *RxReady* interrupt.
- In Packet mode, the receiver will start locking its Bit Synchronizer on a minimum or 12 bits of received preamble, before the reception of correct Data, or Sync Word (if enabled) can occur.

6.2.5 Optimized Frequency Hopping Sequences

In a frequency hopping-like application, it is required to turn off the transmitter when hopping from one channel to another, to avoid spectral splatter and obtain the best spectral purity.

- Transmitter hop from Ch A to Ch B - it is advised to step through the RX mode:
    1. Transceiver is in TX mode in Ch A
    2. Program the MKW01Z128 in RX mode
    3. Change the carrier frequency in the *RegFrf* registers
    4. Turn the transceiver back to TX mode
    5. Respect the TX start procedure
- Receiver hop from Ch A to Ch B -
    1. Transceiver is in RX mode in Ch A
    2. Change the carrier frequency in the *RegFrf* registers Program the transceiver in FS mode
    3. Program the MKW01Z128 in FS mode
    4. Turn the transceiver back to RX mode
    5. Respect the RX start procedure

NOTE

All sequences described above are assuming that the sequencer is turned on (SequencerOff=0 in RegOpMode).

6.3 Listen Mode

The receiver can be set to Listen mode, by setting *ListenOn* in *RegOpMode* to 1 while in Standby mode. In this mode, transceiver spends most of the time in Idle mode, during which only the RC oscillator runs. Periodically the receiver is awakened and listens for an RF signal. If a wanted signal is detected, the receiver is kept on and the data is demodulated.

Otherwise, if a wanted signal hasn't been detected after a pre-defined period of time, the receiver is disabled until the next time period.

This periodical RX wake-up requirement is very common in low power applications. On the transceiver it is handled locally by the Listen mode block without using the MCU resources.

The simplified timing diagram of this procedure is illustrated in Figure 6-5.

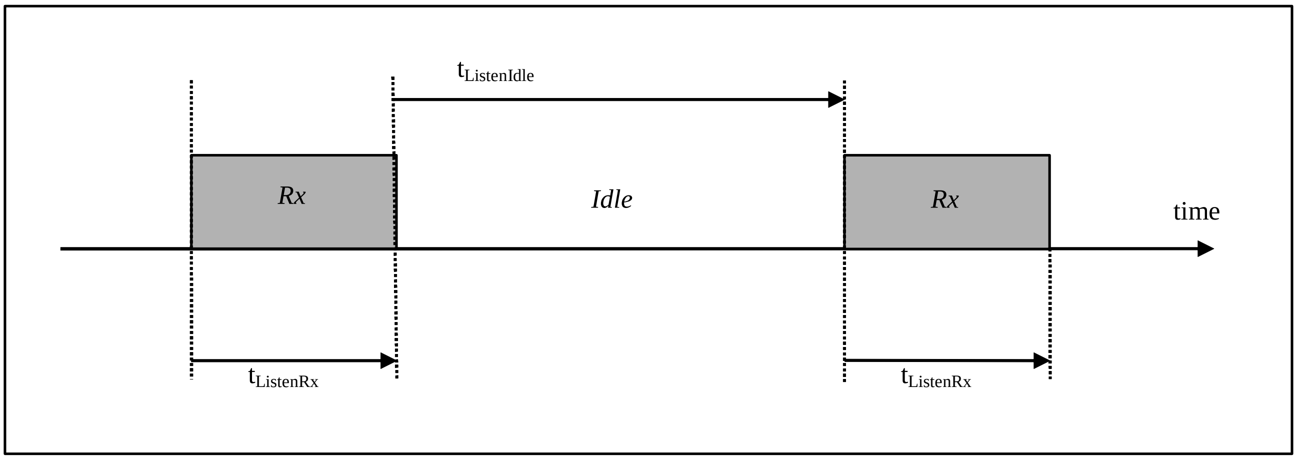


Figure 6-5. Listen Mode Sequence (no wanted signal is received)

6.3.1 Timing

The duration of the Idle phase is given by $t_{ListenIdle}$. The time during which the receiver is on and waits for a signal is given by $t_{ListenRx}$. $t_{ListenRx}$ includes the wake-up time of the receiver, described in section 6.2.3. This duration can be programmed in the configuration registers via the SPI.

Both time periods $t_{ListenRx}$ and $t_{ListenIdle}$ (denoted $t_{ListenX}$ in the following text) are fixed by two parameters from the configuration register and are calculated as follows

:

$$t_{ListenX} = ListenCoefX \cdot ListenResolX$$

where *ListenResolX* is the RX or Idle resolution and is independently programmable on three values (64us, 4.1ms or 262ms), whereas *ListenCoefX* is an integer between 1 and 255. All parameters are located in *RegListen* registers.

The timing ranges are tabulated in Table 6-2 below.

Table 6-2. Range of Durations in Listen Mode

ListenResolX	Min duration (*ListenCoef* = 1)	Max duration (*ListenCoef* = 255)
01	64 us	16 ms
10	4.1 ms	1.04 s
11	0.26 s	67 s

NOTE

- The accuracy of the typical timings given in Table 6-2 will depend in the RC oscillator calibration
- RC oscillator calibration is required, and must be performed at power up. See Section 6.3.4, “RC Timer Accuracy” for details

6.3.2 Criteria

The criteria taken for detecting a wanted signal and hence deciding to maintain the receiver on is defined by *ListenCriteria* in *RegListen1*.

Table 6-3. Signal Acceptance Criteria in Listen Mode

ListenCriteria	Input Signal Power >= *RssiThreshold*	SyncAddressMatch
0	Required	Not Required
1	Required	Required

6.3.3 End of Cycle Actions

The action taken after detection of a packet, is defined by *ListenEnd* in *RegListen3*, as described in the table below.

Table 6-4. End of Listen Cycle Actions

ListenEnd	Description
00	Chip stays in RX mode. Listen mode stops and must be disabled.
01	Chip stays in RX mode until *PayloadReady* or *Timeout* interrupt occurs. It then goes to the mode defined by *Mode*. Listen mode stops and must be disabled.
10	Chip stays in RX mode until *PayloadReady* or *Timeout* interrupt occurs. Listen mode then resumes in Idle state. FIFO content is lost at next RX wakeup.

Upon detection of a valid packet, the sequencing is altered, as shown below:

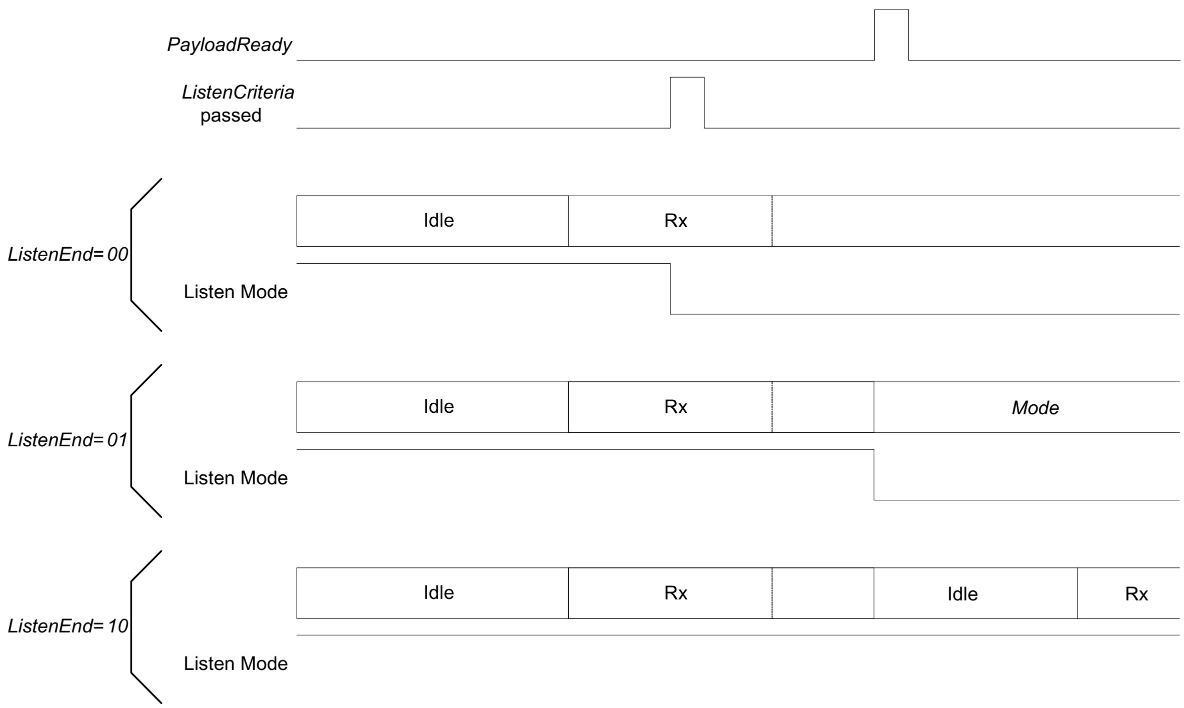


Figure 6-6. Listen Mode Sequence (wanted signal is received)

Listen mode can be disabled by writing *ListenOn* to 0.

6.3.4 RC Timer Accuracy

All timings of the Listen Mode rely on the accuracy of the internal low-power RC oscillator. This oscillator is automatically calibrated at the device power-up, and it is a user-transparent process.

For applications enduring large temperature variations, and for which the power supply is never removed, RC calibration can be performed upon user request. *RcCalStart* in *RegOsc1* can be used to trigger this calibration, and the flag *RcCalDone* will be set automatically when the calibration is over.

6.4 AutoModes

Automatic modes of packet handler can be enabled by configuring the related parameters in *RegAutoModes*. The intermediate mode of the chip is called *IntermediateMode* and the enter and exit conditions to/from this intermediate mode can be configured through the parameters *EnterCondition* & *ExitCondition*. The enter and exit conditions cannot be used independently of each other i.e. both should be enabled at the same time.

The initial and the final state is the one configured in *Mode* in *RegOpMode*. The initial & final states can be different by configuring the modes register while the chip is in intermediate mode. The pictorial description of the auto modes is shown below.

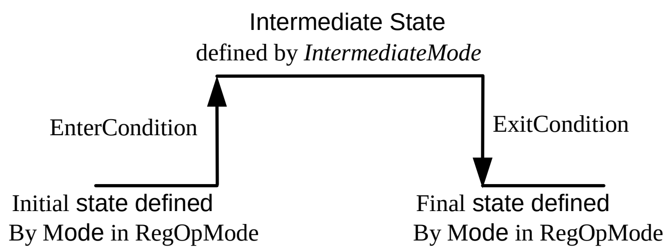


Figure 6-7. Auto Modes of Packet Handler

Some typical examples of AutoModes usage are described below:

- Automatic transmission (AutoTx) : *Mode* = Sleep, *IntermediateMode* = TX, *EnterCondition* = *FifoLevel*, *ExitCondition* = *PacketSent*
- Automatic reception (AutoRx) : *Mode* = RX, *IntermediateMode* = Sleep, *EnterCondition* = *CrcOk*, *ExitCondition* = falling edge of *FifoNotEmpty*
- Automatic reception of acknowledge (AutoRxAck): *Mode* = TX, *IntermediateMode* = RX, *EnterCondition* = *PacketSent*, *ExitCondition* = *CrcOk*

Chapter 7
Transceiver Digital Control and Communications

7.1 Overview

The following figure shows the MKW01Z128 data processing circuit. Its role is to interface the data to/from the modulator/demodulator and the MCU access points (SPI and DIO pins). It also controls all the configuration registers.

The circuit contains several control blocks which are described in the following paragraphs.

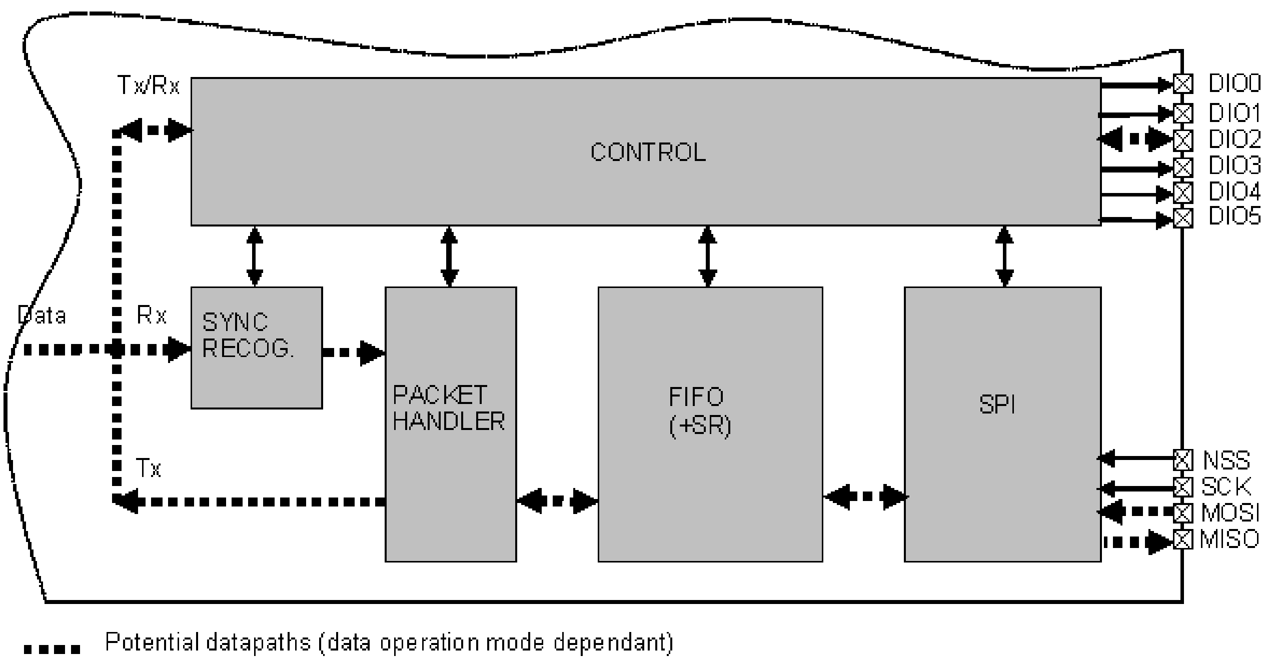


Figure 7-1. MKW01Z128 Data Processing Conceptual View

The MKW01Z128 implements several data operation modes, each with their own data path through the data processing section. Depending on the data operation mode selected, some control blocks are active whilst others remain disabled.

7.1.1 Data Operation Modes

The MKW01Z128 has two different data operation modes selectable by the user:

- Continuous mode: each bit transmitted or received is accessed in real time at the DIO2/DATA pin. This mode may be used if adequate external signal processing is available.

- Packet mode (recommended): user only provides/retrieves payload bytes to/from the FIFO. The packet is automatically built with preamble, Sync word, and optional AES, CRC, and DC-free encoding schemes The reverse operation is performed in reception. The MCU processing overhead is hence significantly reduced compared to Continuous mode. Depending on the optional features activated (CRC, AES, etc) the maximum payload length is limited to FIFO size, 255 bytes or unlimited.

Each of these data operation modes is described fully in the following sections.

7.2 Control Block Description

7.2.1 SPI Interface

The SPI interface gives access to the configuration register via a synchronous full-duplex protocol corresponding to CPOL = 0 and CPHA = 0 in Motorola/Freescale nomenclature. Only the slave side is implemented.

Three access modes to the registers are provided:

- SINGLE access: an address byte followed by a data byte is sent for a write access whereas an address byte is sent and a read byte is received for the read access. The NSS pin goes low at the begin of the frame and goes high after the data byte.
- BURST access: the address byte is followed by several data bytes. The address is automatically incremented internally between each data byte. This mode is available for both read and write accesses. The NSS pin goes low at the beginning of the frame and stay low between each byte. It goes high only after the last byte transfer.
- FIFO access: if the address byte corresponds to the address of the FIFO, then succeeding data byte will address the FIFO. The address is not automatically incremented but is memorized and does not need to be sent between each data byte. The NSS pin goes low at the beginning of the frame and stay low between each byte. It goes high only after the last byte transfer.

Figure below shows a typical SPI single access to a register.

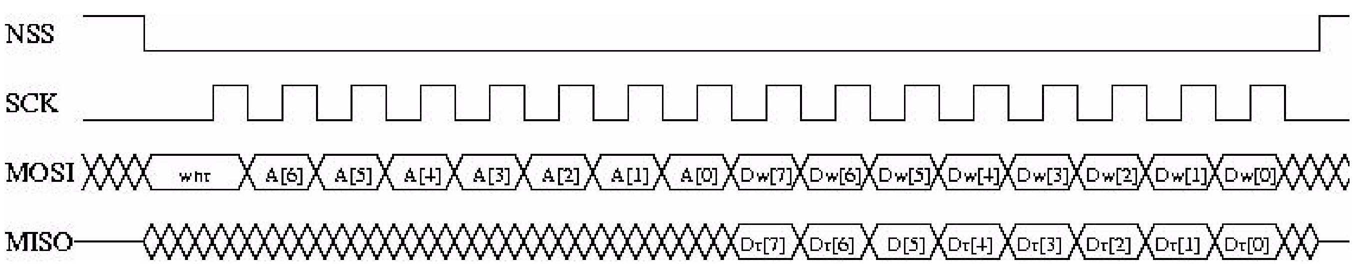


Figure 7-2. SPI Timing Diagram (single access)

MOSI is generated by the master on the falling edge of SCK and is sampled by the slave (i.e. this SPI interface) on the rising edge of SCK. MISO is generated by the slave on the falling edge of SCK.

A transfer always starts by the NSS pin going low. MISO is high impedance when NSS is high.

The first byte is the address byte. It is made of:

- wnr bit, which is 1 for write access and 0 for read access
- 7 bits of address, MSB first

The second byte is a data byte, either sent on MOSI by the master in case of a write access, or received by the master on MISO in case of read access. The data byte is transmitted MSB first.

Proceeding bytes may be sent on MOSI (for write access) or received on MISO (for read access) without rising NSS and re-sending the address. In FIFO mode, if the address was the FIFO address then the bytes will be written / read at the FIFO address. In Burst mode, if the address was not the FIFO address, then it is automatically incremented at each new byte received.

The frame ends when NSS goes high. The next frame must start with an address byte. The SINGLE access mode is actually a special case of FIFO / BURST mode with only 1 data byte transferred.

During the write access, the byte transferred from the slave to the master on the MISO line is the value of the written register before the write operation.

7.2.2 FIFO

7.2.2.1 Overview and Shift Register (SR)

In packet mode of operation, both data to be transmitted and that has been received are stored in a configurable FIFO (First In First Out) device. It is accessed via the SPI interface and provides several interrupts for transfer management.

The FIFO is 1 byte wide hence it only performs byte (parallel) operations, whereas the demodulator functions serially. A shift register is therefore employed to interface the two devices. In transmit mode it takes bytes from the FIFO and outputs them serially (MSB first) at the programmed bit rate to the modulator. Similarly, in RX the shift register gets bit by bit data from the demodulator and writes them byte by byte to the FIFO. This is illustrated in figure below.

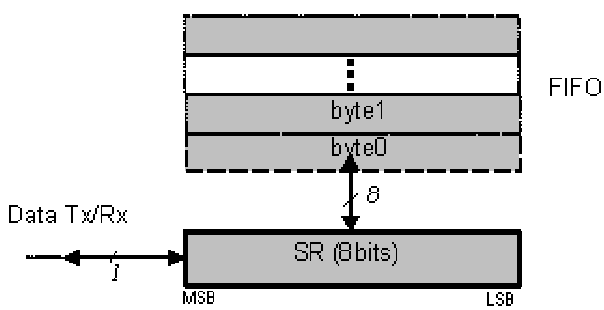


Figure 7-3. FIFO and Shift Register (SR)

NOTE

When switching to Sleep mode, the FIFO can only be used once the ModeReady flag is set (quasi immediate from all modes except from TX)

7.2.2.2 Size

The FIFO size is fixed to 66 bytes.

7.2.2.3 Interrupt Sources and Flags

- *FifoNotEmpty*: *FifoNotEmpty* interrupt source is low when byte 0, i.e. whole FIFO, is empty. Otherwise it is high. Note that when retrieving data from the FIFO, *FifoNotEmpty* is updated on NSS falling edge, i.e. when *FifoNotEmpty* is updated to low state the currently started read operation must be completed. In other words, *FifoNotEmpty* state must be checked after each read operation for a decision on the next one (*FifoNotEmpty* = 1: more byte(s) to read; *FifoNotEmpty* = 0: no more byte to read).
- *FifoFull*: *FifoFull* interrupt source is high when the last FIFO byte, i.e. the whole FIFO, is full. Otherwise it is low.
- *FifoOverrunFlag*: *FifoOverrunFlag* is set when a new byte is written by the user (in TX or Standby modes) or the SR (in RX mode) while the FIFO is already full. Data is lost and the flag should be cleared by writing a 1, note that the FIFO will also be cleared.
- *PacketSent*: *PacketSent* interrupt source goes high when the SR's last bit has been sent.
- *FifoLevel*: Threshold can be programmed by *FifoThreshold* in *RegFifoThresh*. Its behavior is illustrated in figure below.

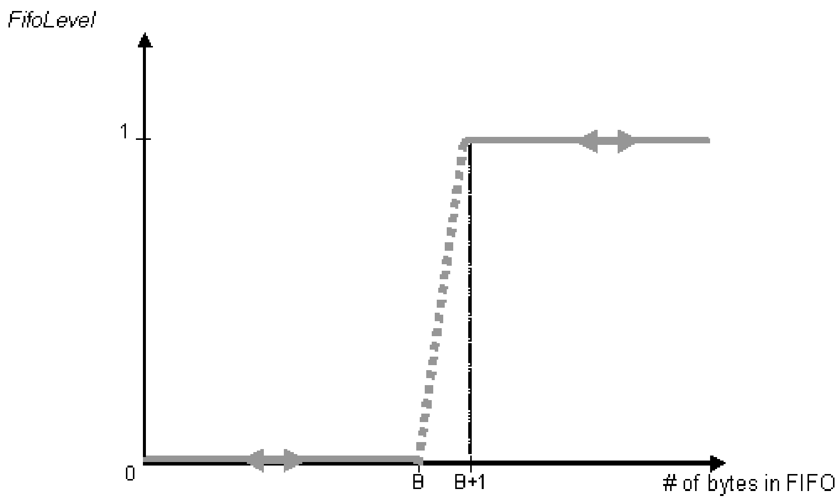


Figure 7-4. FifoLevel IRQ Source Behavior

NOTE

FifoLevel interrupt is updated only after a read or write operation on the FIFO. Thus the interrupt cannot be dynamically updated by only changing the FifoThreshold parameter.

FifoLevel interrupt is valid as long as FifoFull does not occur. An empty FIFO will restore its normal operation

7.2.2.4 FIFO Clearing

Table below summarizes the status of the FIFO when switching between different modes.

Table 7-1. Status of FIFO when Switching Between Different Modes of the Chip

From	To	FIFO status	Comments
Stdby	Sleep	Not cleared	
Sleep	Stdby	Not cleared	
Stdby/Sleep	TX	Not cleared	To allow the user to write the FIFO in Stdby/Sleep before TX
Stdby/Sleep	RX	Cleared	
RX	TX	Cleared	
RX	Stdby/Sleep	Not cleared	To allow the user to read FIFO in Stdby/Sleep mode after RX
TX	Any	Cleared	

7.2.3 Sync Word Recognition

7.2.3.1 Overview

Sync word recognition (also called Pattern recognition) is activated by setting *SyncOn* in *RegSyncConfig*. The bit synchronizer must also be activated in continuous mode (automatically done in Packet mode) .

The block behaves like a shift register; it continuously compares the incoming data with its internally programmed Sync word and sets *SyncAddressMatch* when a match is detected. This is illustrated in Figure 7-5 below.

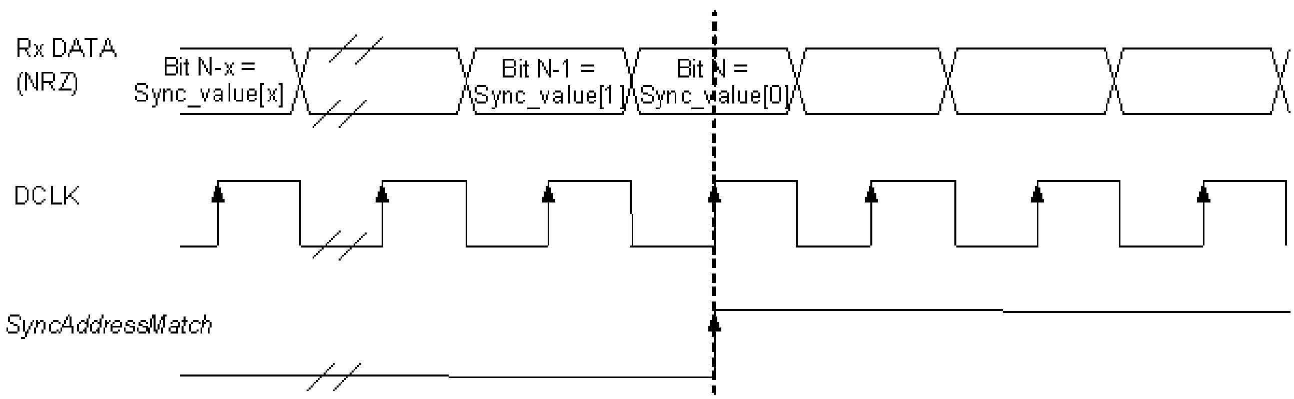


Figure 7-5. Sync Word Recognition

During the comparison of the demodulated data, the first bit received is compared with bit 7 (MSB) of *RegSyncValue1* and the last bit received is compared with bit 0 (LSB) of the last byte whose address is determined by the length of the Sync word.

When the programmed Sync word is detected the user can assume that this incoming packet is for the node and can be processed accordingly.

SyncAddressMatch is cleared when leaving RX or FIFO is emptied.

7.2.3.2 Configuration

- Size: Sync word size can be set from 1 to 8 bytes (i.e. 8 to 64 bits) via *SyncSize* in *RegSyncConfig*. In Packet mode this field is also used for Sync word generation in TX mode.
- Error tolerance: The number of errors tolerated in the Sync word recognition can be set from 0 to 7 bits to via *SyncTol*.
- Value: The Sync word value is configured in *SyncValue(63:0)*. In Packet mode this field is also used for Sync word generation in TX mode.

NOTE

SyncValue choices containing 0x00 bytes are not allowed.

7.2.4 Packet Handler

The packet handler is the block used in Packet mode. Its functionality is fully described in section 7.5.

7.2.5 Control

The control block configures and controls the full chip's behavior according to the settings programmed in the configuration registers.

7.3 Digital IO Pins Mapping

Six general purpose IO pins are available on the MKW01Z128, and their configuration in Continuous or Packet mode is controlled through *RegDioMapping1* and *RegDioMapping2*.

7.3.1 DIO Pins Mapping in Continuous Mode

Table 7-2. DIO Mapping, Continuous Mode

Mode	Diox Mapping	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
Sleep	00	-	-	-	-	-	-
	01	-	-	-	-	-	-
	10	LowBat	LowBat	AutoMode	-	LowBat	LowBat
	11	ModeReady	-	-	-	-	ModeReady
Stdby	00	ClkOut	-	-	-	-	-
	01	-	-	-	-	-	-
	10	LowBat	LowBat	AutoMode	-	LowBat	LowBat
	11	ModeReady	-	-	-	-	ModeReady
FS	00	ClkOut	-	-	-	-	PllLock
	01	-	-	-	-	-	-
	10	LowBat	LowBat	AutoMode	-	LowBat	LowBat
	11	ModeReady	PllLock	-	-	PllLock	ModeReady
Rx	00	ClkOut	Timeout	Rssi	Data	Dclk	SyncAddress
	01	Rssi	RxReady	RxReady	Data	RxReady	Timeout
	10	LowBat	SyncAddress	AutoMode	Data	LowBat	Rssi
	11	ModeReady	PllLock	Timeout	Data	SyncAddress	ModeReady
Tx	00	ClkOut	TxReady	TxReady	Data	Dclk	PllLock
	01	ClkOut	TxReady	TxReady	Data	TxReady	TxReady
	10	LowBat	LowBat	AutoMode	Data	LowBat	LowBat
	11	ModeReady	PllLock	TxReady	Data	PllLock	ModeReady

7.3.2 DIO Pins Mapping in Packet Mode

Table 7-3. DIO Mapping, Packet Mode

Mode	Diox Mapping	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
Sleep	00	-	-	FifoFull	FifoNotEmpty	FifoLevel	-
	01	-	-	-	-	FifoFull	-
	10	LowBat	LowBat	LowBat	LowBat	FifoNotEmpty	LowBat
	11	ModeReady	-	-	AutoMode	-	-
Stdby	00	ClkOut	-	FifoFull	FifoNotEmpty	FifoLevel	-
	01	-	-	-	-	FifoFull	-
	10	LowBat	LowBat	LowBat	LowBat	FifoNotEmpty	LowBat
	11	ModeReady	-	-	AutoMode	-	-
FS	00	ClkOut	-	FifoFull	FifoNotEmpty	FifoLevel	-
	01	-	-	-	-	FifoFull	-
	10	LowBat	LowBat	LowBat	LowBat	FifoNotEmpty	LowBat
	11	ModeReady	PllLock	PllLock	AutoMode	PllLock	PllLock
Rx	00	ClkOut	Timeout	FifoFull	FifoNotEmpty	FifoLevel	CrcOk
	01	Data	Rssi	Rssi	Data	FifoFull	PayloadReady
	10	LowBat	RxReady	SyncAddress	LowBat	FifoNotEmpty	SyncAddress
	11	ModeReady	PllLock	PllLock	AutoMode	Timeout	Rssi
Tx	00	ClkOut	ModeReady	FifoFull	FifoNotEmpty	FifoLevel	PacketSent
	01	Data	TxReady	TxReady	Data	FifoFull	TxReady
	10	LowBat	LowBat	LowBat	LowBat	FifoNotEmpty	LowBat
	11	ModeReady	PllLock	PllLock	AutoMode	PllLock	PllLock

NOTE

Received Data is only shown on the Data signal between RxReady and PayloadReady's rising edges

7.4 Continuous Mode

7.4.1 General Description

As illustrated in Figure 7-6, in Continuous mode the NRZ data to (from) the (de)modulator is directly accessed by the MCU on the bidirectional DIO2/DATA pin. The FIFO and packet handler are thus inactive.

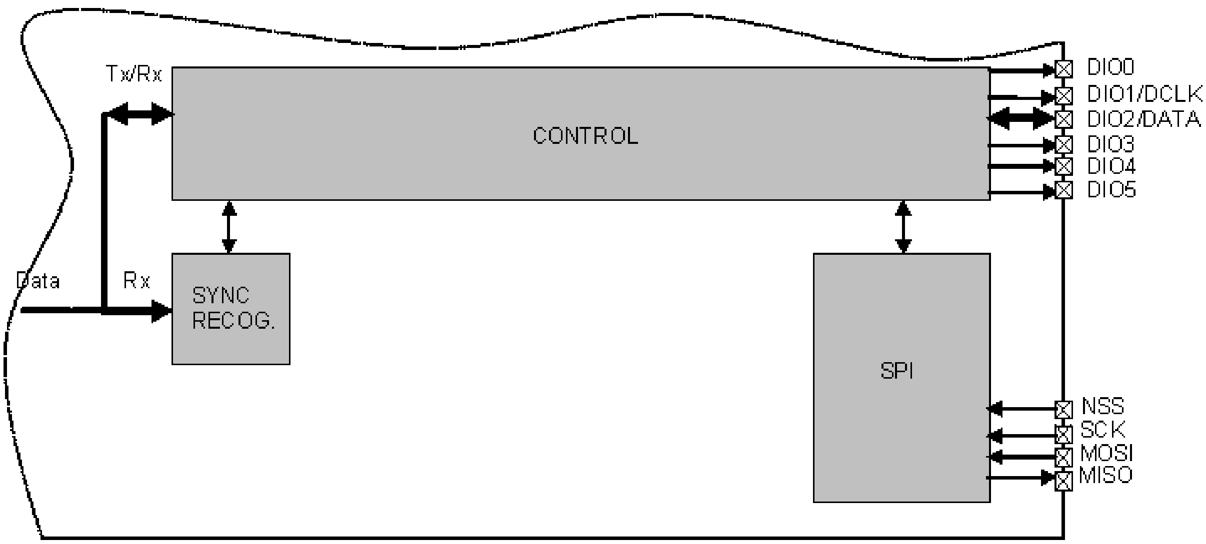


Figure 7-6. Continuous Mode Conceptual View

7.4.2 TX Processing

In TX mode, a synchronous data clock for an external MCU is provided on DIO1/DCLK pin. Clock timing with respect to the data is illustrated in Figure 7-7. DATA is internally sampled on the rising edge of DCLK so the MCU can change logic state anytime outside the grayed out setup/hold zone.

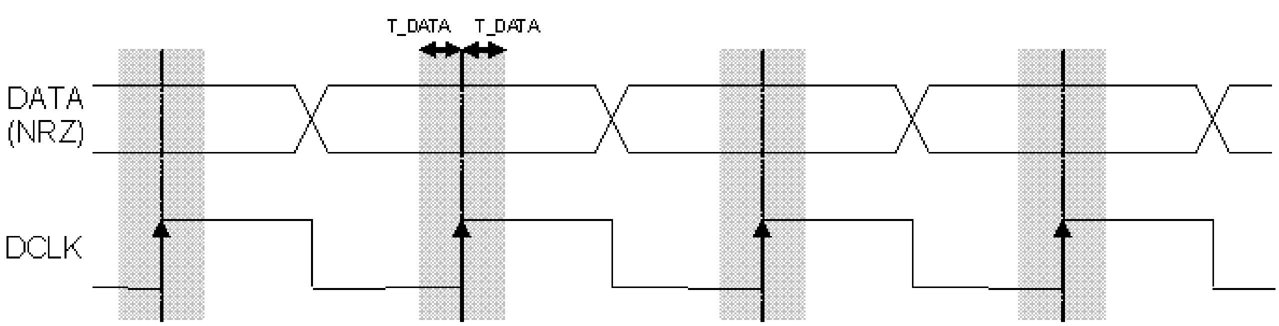


Figure 7-7. TX Processing in Continuous Mode

NOTE

The use of DCLK is required when the modulation shaping is enabled.

7.4.3 RX Processing

If the bit synchronizer is disabled, the raw demodulator output is made directly available on DATA pin and no DCLK signal is provided.

Conversely, if the bit synchronizer is enabled, synchronous cleaned data and clock are made available respectively on DIO2/DATA and DIO1/DCLK pins. DATA is sampled on the rising edge of DCLK and updated on the falling edge as illustrated below.

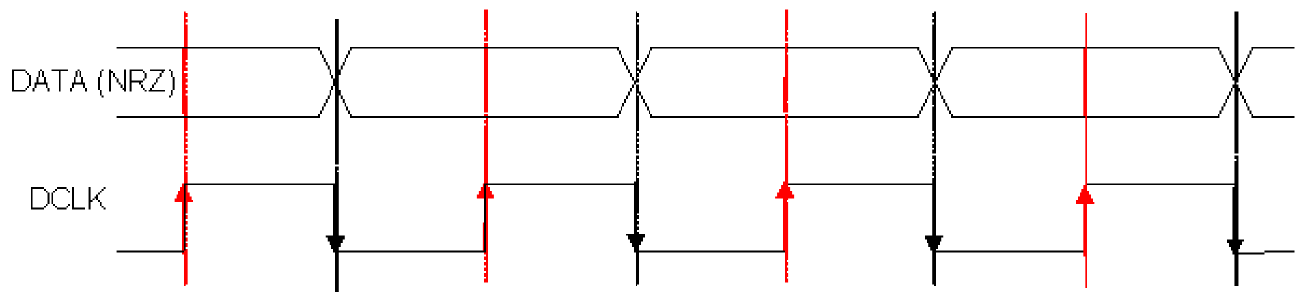


Figure 7-8. RX Processing in Continuous Mode

NOTE

In Continuous mode it is always recommended to enable the bit synchronizer to clean the DATA signal even if the DCLK signal is not used by the MCU (bit synchronizer is automatically enabled in Packet mode).

7.5 Packet Mode

7.5.1 General Description

In Packet mode the NRZ data to (from) the (de)modulator is not directly accessed by the MCU but stored in the FIFO and accessed via the SPI interface.

In addition, the MKW01Z128 packet handler performs several packet oriented tasks such as Preamble and Sync word generation, CRC calculation/check, whitening/dewhitening of data, Manchester encoding/decoding, address filtering, AES encryption/decryption, etc. This simplifies software and reduces MCU overhead by performing these repetitive tasks within the RF chip itself.

Another important feature is ability to fill and empty the FIFO in Sleep/Stdby mode, ensuring optimum power consumption and adding more flexibility for the software.

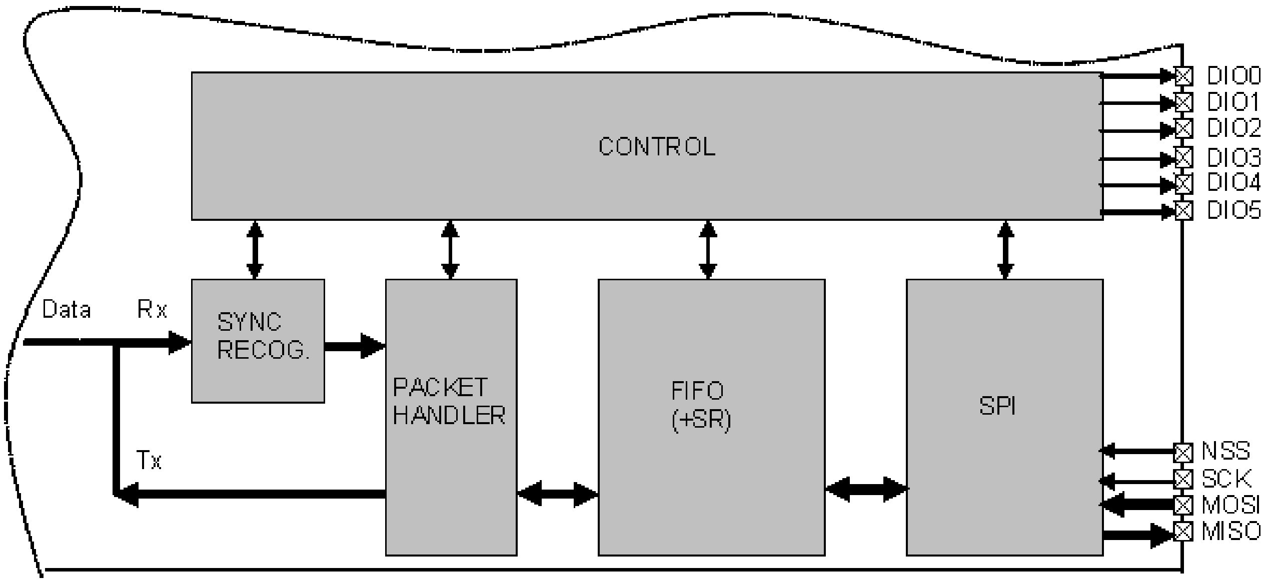


Figure 7-9. Packet Mode Conceptual View

NOTE

The Bit Synchronizer is automatically enabled in Packet mode.

7.5.2 Packet Format

7.5.2.1 Fixed Length Packet Format

Fixed length packet format is selected when bit *PacketFormat* is set to 0 and *PayloadLength* is set to any value greater than 0.

In applications where the packet length is fixed in advance, this mode of operation may be of interest to minimize RF overhead (no length byte field is required). All nodes, whether TX only, RX only, or TX/RX should be programmed with the same packet length value.

The length of the payload is limited to 255 bytes if AES is not enabled else the message is limited to 64 bytes (i.e. max 65 bytes payload if Address byte is enabled).

The length programmed in *PayloadLength* relates only to the payload which includes the message and the optional address byte. In this mode, the payload must contain at least one byte, i.e. address or message byte.

An illustration of a fixed length packet is shown below. It contains the following fields:

- Preamble (1010...)
- Sync word (Network ID)
- Optional Address byte (Node ID)
- Message data
- Optional 2-bytes CRC checksum

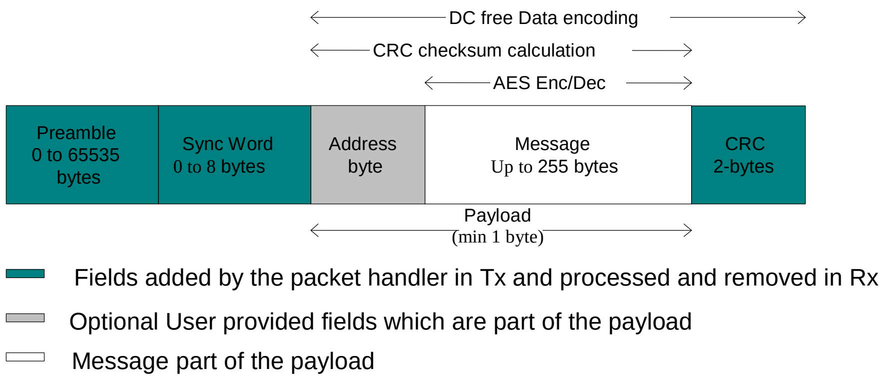


Figure 7-10. Fixed Length Packet Format

7.5.2.2 Variable Length Packet Format

Variable length packet format is selected when bit *PacketFormat* is set to 1.

This mode is useful in applications where the length of the packet is not known in advance and can vary over time. It is then necessary for the transmitter to send the length information together with each packet in order for the receiver to operate properly.

In this mode the length of the payload, indicated by the length byte, is given by the first byte of the FIFO and is limited to 255 bytes if AES is not enabled else the message is limited to 64 bytes (i.e. max 66 bytes payload if Address byte is enabled). Note that the length byte itself is not included in its calculation. In this mode, the payload must contain at least 2 bytes, i.e. length + address or message byte.

An illustration of a variable length packet is shown below. It contains the following fields:

- Preamble (1010...)
- Sync word (Network ID)
- Length byte
- Optional Address byte (Node ID)
- Message data
- Optional 2-bytes CRC checksum

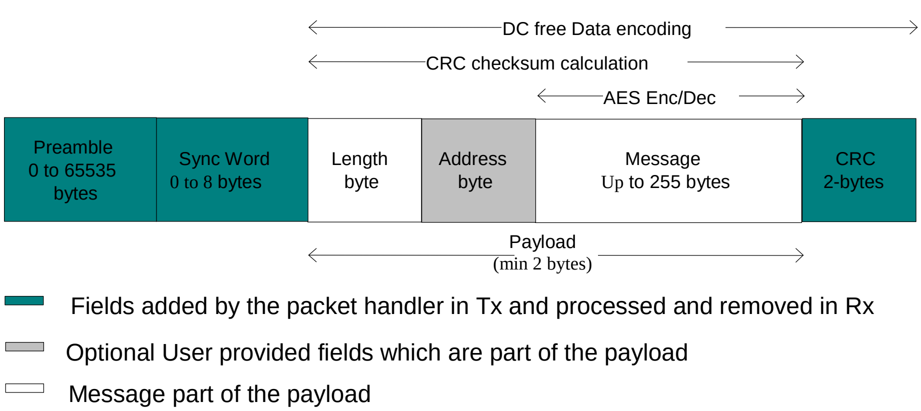


Figure 7-11. Variable Length Packet Format

7.5.2.3 Unlimited Length Packet Format

Unlimited length packet format is selected when bit *PacketFormat* is set to 0 and *PayloadLength* is set to 0.

The user can then transmit and receive packet of arbitrary length and *PayloadLength* register is not used in TX/RX modes for counting the length of the bytes transmitted/received.

In TX the data is transmitted depending on the *TxStartCondition* bit. On the RX side the data processing features like Address filtering, Manchester encoding and data whitening are not available if the sync pattern length is set to zero (*SyncOn = 0*). The filling of the FIFO in this case can be controlled by the bit *FifoFillCondition*. The CRC detection in RX is also not supported in this mode of the packet handler, however CRC generation in TX is operational. The interrupts like *CrcOk* & *PayloadReady* are not available either.

An unlimited length packet is made up of the following fields:

- Preamble (1010...).
- Sync word (Network ID).
- Optional Address byte (Node ID).
- Message data
- Optional 2-bytes CRC checksum (TX only)

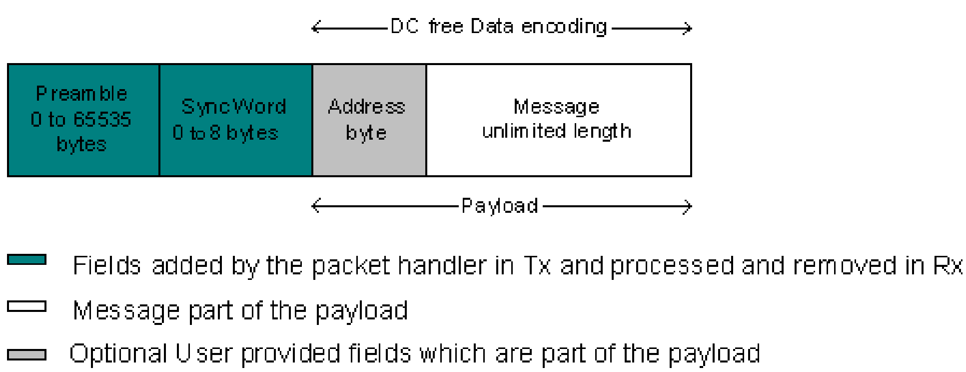


Figure 7-12. Unlimited Length Packet Format

7.5.3 TX Processing (without AES)

In TX mode the packet handler dynamically builds the packet by performing the following operations on the payload available in the FIFO:

- Add a programmable number of preamble bytes
- Add a programmable Sync word
- Optionally calculating CRC over complete payload field (optional length byte + optional address byte + message) and appending the 2 bytes checksum.
- Optional DC-free encoding of the data (Manchester or whitening)

Only the payload (including optional address and length fields) is required to be provided by the user in the FIFO.

The transmission of packet data is initiated by the Packet Handler only if the chip is in TX mode and the transmission condition defined by *TxStartCondition* is fulfilled. If transmission condition is not fulfilled then the packet handler transmits a preamble sequence until the condition is met. This happens only if the preamble length /= 0, otherwise it transmits a zero or one until the condition is met to transmit the packet data.

The transmission condition itself is defined as:

- if *TxStartCondition* = 1, the packet handler waits until the first byte is written into the FIFO, then it starts sending the preamble followed by the sync word and user payload
- If *TxStartCondition* = 0, the packet handler waits until the number of bytes written in the FIFO is equal to the number defined in *RegFifoThresh* + 1
- If the condition for transmission was already fulfilled i.e. the FIFO was filled in Sleep/Stdby then the transmission of packet starts immediately on enabling TX

7.5.4 RX Processing (without AES)

In RX mode the packet handler extracts the user payload to the FIFO by performing the following operations:

- Receiving the preamble and stripping it off
- Detecting the Sync word and stripping it off
- Optional DC-free decoding of data
- Optionally checking the address byte
- Optionally checking CRC and reflecting the result on *CrcOk.*

Only the payload (including optional address and length fields) is made available in the FIFO.

When the RX mode is enabled the demodulator receives the preamble followed by the detection of sync word. If fixed length packet format is enabled then the number of bytes received as the payload is given by the *PayloadLength* parameter.

In variable length mode the first byte received after the sync word is interpreted as the length of the received packet. The internal length counter is initialized to this received length. The *PayloadLength* register is set to a value which is greater than the maximum expected length of the received packet. If the received length is greater than the maximum length stored in *PayloadLength* register the packet is discarded otherwise the complete packet is received.

If the address check is enabled then the second byte received in case of variable length and first byte in case of fixed length is the address byte. If the address matches to the one in the *NodeAddress* field, reception of the data continues otherwise it's stopped. The CRC check is performed if *CrcOn* = 1 and the result is available in *CrcOk* indicating that the CRC was successful. An interrupt (*PayloadReady*) is also generated on DIO0 as soon as the payload is available in the FIFO. The payload available in the FIFO can also be read in Sleep/Standby mode.

If the CRC fails the *PayloadReady* interrupt is not generated and the FIFO is cleared. This function can be overridden by setting *CrcAutoClearOff* = 1, forcing the availability of *PayloadReady* interrupt and the payload in the FIFO even if the CRC fails.

7.5.5 AES

AES is the symmetric-key block cipher that provides the cryptographic capabilities to the transceiver. The system proposed can work with 128-bit long fixed keys. The fixed key is stored in a 16-byte write only user configuration register, which retains its value in Sleep mode.

As shown in Figure 7-10 and Figure 7-11 above the message part of the Packet can be encrypted and decrypted with the cipher 128- cipher key stored in the configuration registers.

7.5.5.1 TX Processing

1. User enters the data to be transmitted in FIFO in Stdby/Sleep mode and gives the transmit command.
2. On TX command the Packet handler state machine takes over the control and If encryption is enabled then the message inside the FIFO is read in blocks of 16 bytes (padded with 0s if needed), encrypted and stored back to FIFO. All this processing is done in TX mode before enabling the packet handling state machine. Only the Message part of the packet is encrypted and preamble, sync word, length byte, address byte and CRC are not encrypted.
3. Once the encryption is done the Packet handling state machine is enabled to transmit the data.

7.5.5.2 RX Processing

1. The data received is stored in the FIFO, The address, CRC interrupts are generated as usual because these parameters were not encrypted.
2. Once the complete packet has been received. The data is read from the FIFO, decrypted and written back to FIFO. The *PayloadReady* interrupt is issued once the decrypted data is ready in the FIFO for reading via the SPI interface.

The AES encryption/decryption cannot be used on the fly i.e. while transmitting and receiving data. Thus when AES encryption/decryption is enabled, the FIFO acts as a simple buffer. This buffer is filled before initiating any transmission. The data in the buffer is then encrypted before the transmission can begin. On the receive side the decryption is initiated only once the complete packet has been received in the buffer.

The encryption/decryption process takes approximately 7.0 us per 16-byte block. Thus for a maximum of 4 blocks (i.e. 64 bytes) it can take up to 28 us for completing the cryptographic operations.

The receive side sees the AES decryption time as a sequential delay before the *PayloadReady* interrupt is available.

The TX side sees the AES encryption time as a sequential delay in the startup of the TX chain, thus the startup time of the TX will increase according to the length of data.

In Fixed length mode the Message part of the payload that can be encrypted/decrypted can be 64 bytes long. If the address filtering is enabled, the length of the payload should be at max 65 bytes in this case.

In Variable length mode the Max message size that can be encrypted/decrypted is also 64 bytes when address filtering is disabled, else it is 48 bytes. Thus, including length byte, the length of the payload is max 65 or 50 bytes (the latter when address filtering is enabled).

If the address filtering is expected then *AddressFiltering* must be enabled on the transmitter side as well to prevent address byte to be encrypted.

Crc check being performed on encrypted data, *CrcOk* interrupt will occur "decryption time" before *PayloadReady* interrupt.

7.5.6 Handling Large Packets

When Payload length exceeds FIFO size (66 bytes) whether in fixed, variable or unlimited length packet format, in addition to *PacketSent* in TX and *PayloadReady* or *CrcOk* in RX, the FIFO interrupts/flags can be used as described below:

- For TX:
  — FIFO can be prefilled in Sleep/Standby but must be refilled "on-the-fly" during TX with the rest of the payload.

1. Prefill FIFO (in Sleep/Standby first or directly in TX mode) until *FifoThreshold* or *FifoFull* is set
2. In TX, wait for *FifoThreshold* or *FifoNotEmpty* to be cleared (i.e. FIFO is nearly empty)
3. Write bytes into the FIFO until *FifoThreshold* or *FifoFull* is set.
4. Continue to step 2 until the entire message has been written to the FIFO (*PacketSent* will fire when the last bit of the packet has been sent).

- For RX:
  — FIFO must be unfilled "on-the-fly" during RX to prevent FIFO overrun.

1. Start reading bytes from the FIFO when *FifoNotEmpty* or *FifoThreshold* becomes set.
2. Suspend reading from the FIFO if *FifoNotEmpty* clears before all bytes of the message have been read
3. Continue to step 1 until *PayloadReady* or *CrcOk* fires
4. Read all remaining bytes from the FIFO either in RX or Sleep/Standby mode

NOTE

AES encryption is not feasible on large packets, since all Payload bytes need to be in the FIFO at the same time to perform encryption.

7.5.7 Packet Filtering

MKW01Z128's packet handler offers several mechanisms for packet filtering, ensuring that only useful packets are made available to the MCU, reducing significantly system power consumption and software complexity.

7.5.7.1 Sync Word Based

Sync word filtering/recognition is used for identifying the start of the payload and also for network identification. As previously described, the Sync word recognition block is configured (size, error tolerance, value) in *RegSyncValue* registers. This information is used, both for appending Sync word in TX, and filtering packets in RX.

Every received packet which does not start with this locally configured Sync word is automatically discarded and no interrupt is generated.

When the Sync word is detected, payload reception automatically starts and *SyncAddressMatch* is asserted.

NOTE

Sync Word values containing 0x00 byte(s) are forbidden.

7.5.7.2 Address Based

Address filtering can be enabled via the *AddressFiltering* bits. It adds another level of filtering, above Sync word (i.e. Sync must match first), typically useful in a multi-node networks where a network ID is shared between all nodes (Sync word) and each node has its own ID (address).

Two address based filtering options are available:

- *AddressFiltering = 01*: Received address field is compared with internal register *NodeAddress*. If they match then the packet is accepted and processed, otherwise it is discarded.
- *AddressFiltering = 10*: Received address field is compared with internal registers *NodeAddress* and *BroadcastAddress*. If either is a match, the received packet is accepted and processed, otherwise it is discarded. This additional check with a constant is useful for implementing broadcast in a multi-node networks

Please note that the received address byte, as part of the payload, is not stripped off the packet and is made available in the FIFO. In addition, *NodeAddress* and *AddressFiltering* only apply to RX. On TX side, if address filtering is expected, the address byte should simply be put into the FIFO like any other byte of the payload.

As address filtering requires a Sync word match, both features share the same interrupt flag *SyncAddressMatch*.

7.5.7.3 Length Based

In variable length Packet mode, *PayloadLength* must be programmed with the maximum payload length permitted. If received length byte is smaller than this maximum then the packet is accepted and processed, otherwise it is discarded.

Please note that the received length byte, as part of the payload, is not stripped off the packet and is made available in the FIFO.

To disable this function the user should set the value of the *PayloadLength* to 255.

7.5.7.4 CRC Based

The CRC check is enabled by setting bit *CrcOn* in *RegPacketConfig1*. It is used for checking the integrity of the message.

- On TX side a two byte CRC checksum is calculated on the payload part of the packet and appended to the end of the message

- On RX side the checksum is calculated on the received payload and compared with the two checksum bytes received. The result of the comparison is stored in bit *CrcOk*.

By default, if the CRC check fails then the FIFO is automatically cleared and no interrupt is generated. This filtering function can be disabled via *CrcAutoClearOff* bit and in this case, even if CRC fails, the FIFO is not cleared and only *PayloadReady* interrupt goes high. Please note that in both cases, the two CRC checksum bytes are stripped off by the packet handler and only the payload is made available in the FIFO.

The CRC is based on the CCITT polynomial as shown below. This implementation also detects errors due to leading and trailing zeros.

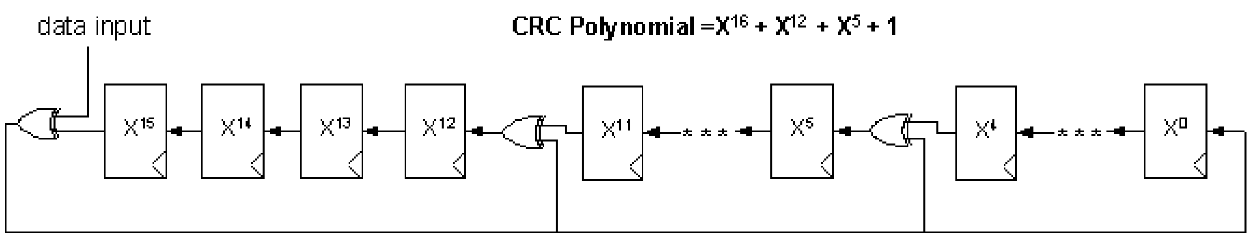


Figure 7-13. CRC Implementation

7.5.8 DC-Free Data Mechanisms

The payload to be transmitted may contain long sequences of 1's and 0's, which introduces a DC bias in the transmitted signal. The radio signal thus produced has a non uniform power distribution over the occupied channel bandwidth. It also introduces data dependencies in the normal operation of the demodulator. Thus it is useful if the transmitted data is random and DC free.

For such purposes, two techniques are made available in the packet handler: Manchester encoding and data whitening.

NOTE

Only one of the two methods should be enabled at a time.

7.5.8.1 Manchester Encoding

Manchester encoding/decoding is enabled if *DcFree = 01* and can only be used in Packet mode.

The NRZ data is converted to Manchester code by coding '1' as "10" and '0' as "01".

In this case, the maximum chip rate is the maximum bit rate given in the specifications section and the actual bit rate is half the chip rate.

Manchester encoding and decoding is only applied to the payload and CRC checksum while preamble and Sync word are kept NRZ. However, the chip rate from preamble to CRC is the same and defined by *BitRate* in *RegBitRate* (Chip Rate = Bit Rate NRZ = 2 x Bit Rate Manchester).

Manchester encoding/decoding is thus made transparent for the user, who still provides/retrieves NRZ data to/from the FIFO.

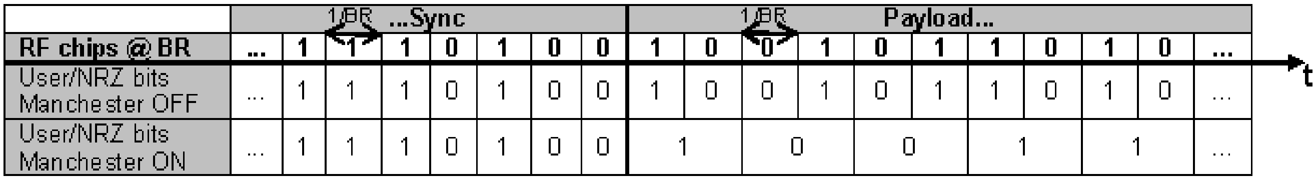


Figure 7-14. Manchester Encoding/Decoding

7.5.8.2 Data Whitening

Another technique called whitening or scrambling is widely used for randomizing the user data before radio transmission. The data is whitened using a random sequence on the TX side and de-whitened on the RX side using the same sequence. Comparing to Manchester technique it has the advantage of keeping NRZ data rate i.e. actual bit rate is not halved.

The whitening/de-whitening process is enabled if *DcFree = 10*. A 9-bit LFSR is used to generate a random sequence. The payload and 2-byte CRC checksum is then XORed with this random sequence as shown below. The data is de-whitened on the receiver side by XORing with the same random sequence.

Payload whitening/de-whitening is thus made transparent for the user, who still provides/retrieves NRZ data to/from the FIFO.

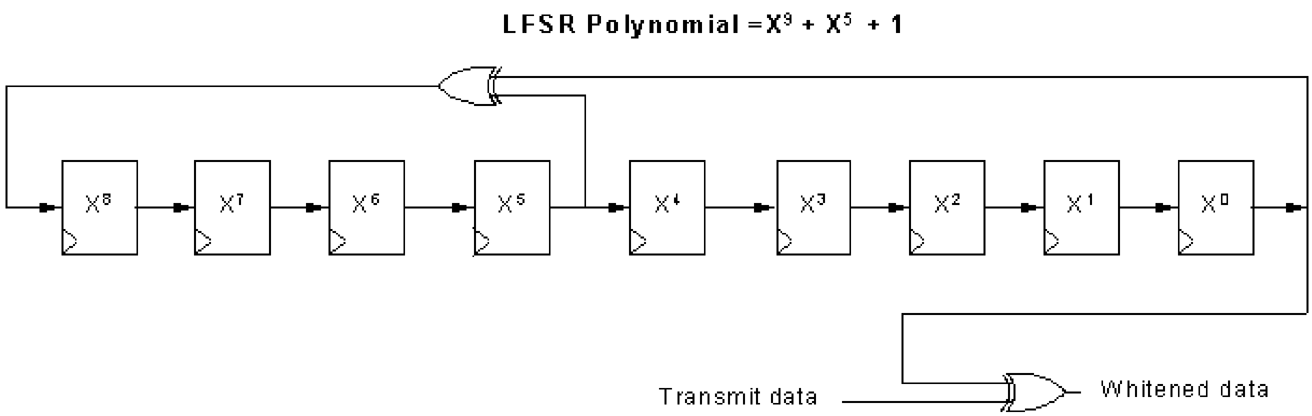


Figure 7-15. Data Whitening

7.6 Register Summary

Table 7-4. Registers Summary

Address	Register Name	Reset (built-in)	Default (recommended)	Description
0x00	RegFifo	0x00		FIFO read/write access
0x01	RegOpMode	0x04		Operating modes of the transceiver
0x02	RegDataModul	0x00		Data operation mode and Modulation settings
0x03	RegBitrateMsb	0x1A		Bit Rate setting, Most Significant Bits
0x04	RegBitrateLsb	0x0B		Bit Rate setting, Least Significant Bits

Table 7-4. Registers Summary

Address	Register Name	Reset (built-in)	Default (recommended)	Description
0x05	RegFdevMsb	0x00		Frequency Deviation setting, Most Significant Bits
0x06	RegFdevLsb	0x52		Frequency Deviation setting, Least Significant Bits
0x07	RegFrfMsb	0xE4		RF Carrier Frequency, Most Significant Bits
0x08	RegFrfMid	0xC0		RF Carrier Frequency, Intermediate Bits
0x09	RegFrfLsb	0x00		RF Carrier Frequency, Least Significant Bits
0x0A	RegOsc1	0x41		RC Oscillators Settings
0x0B	RegAfcCtrl	0x00		AFC control in low modulation index situations
0x0C	RegLowBat	0x02		Low Battery Indicator Settings
0x0D	RegListen1	0x92		Listen Mode settings
0x0E	RegListen2	0xF5		Listen Mode Idle duration
0x0F	RegListen3	0x20		Listen Mode RX duration
0x10	RegVersion	0x22		RF ID relating the silicon revision
0x11	RegPaLevel	0x9F		PA selection and Output Power control
0x12	RegPaRamp	0x09		Control of the PA ramp time in FSK mode
0x13	RegOcp	0x1A		Over Current Protection control
0x14	Reserved14	0x40		-
0x15	Reserved15	0xB0		-
0x16	Reserved16	0x7B		-
0x17	Reserved17	0x9B		-
0x18	RegLna	0x08	0x88	LNA settings
0x19	RegRxBw	0x86	0x55	Channel Filter BW Control
0x1A	RegAfcBw	0x8A	0x8B	Channel Filter BW control during the AFC routine
0x1B	RegOokPeak	0x40		OOK demodulator selection and control in peak mode
0x1C	RegOokAvg	0x80		Average threshold control of the OOK demodulator
0x1D	RegOokFix	0x06		Fixed threshold control of the OOK demodulator
0x1E	RegAfcFei	0x10		AFC and FEI control and status
0x1F	RegAfcMsb	0x00		MSB of the frequency correction of the AFC
0x20	RegAfcLsb	0x00		LSB of the frequency correction of the AFC
0x21	RegFeiMsb	0x00		MSB of the calculated frequency error
0x22	RegFeiLsb	0x00		LSB of the calculated frequency error
0x23	RegRssiConfig	0x02		RSSI-related settings

Table 7-4. Registers Summary

Address	Register Name	Reset (built-in)	Default (recommended)	Description
0x24	RegRssiValue	0xFF		RSSI value in dBm
0x25	RegDioMapping1	0x00		Mapping of pins DIO0 to DIO3
0x26	RegDioMapping2	0x05	0x07	Mapping of pins DIO4 and DIO5, ClkOut frequency
0x27	RegIrqFlags1	0x80		Status register: PLL Lock state, Timeout, RSSI > Threshold...
0x28	RegIrqFlags2	0x00		Status register: FIFO handling flags, Low Battery detection...
0x29	RegRssiThresh	0xFF	0xE4	RSSI Threshold control
0x2A	RegRxTimeout1	0x00		Timeout duration between RX request and RSSI detection
0x2B	RegRxTimeout2	0x00		Timeout duration between RSSI detection and *PayloadReady*
0x2C	RegPreambleMsb	0x00		Preamble length, MSB
0x2D	RegPreambleLsb	0x03		Preamble length, LSB
0x2E	RegSyncConfig	0x98		Sync Word Recognition control
0x2F-0x36	RegSyncValue1-8	0x00	0x01	Sync Word bytes, 1 through 8
0x37	RegPacketConfig1	0x10		Packet mode settings
0x38	RegPayloadLength	0x40		Payload length setting
0x39	RegNodeAdrs	0x00		Node address
0x3A	RegBroadcastAdrs	0x00		Broadcast address
0x3B	RegAutoModes	0x00		Auto modes settings
0x3C	RegFifoThresh	0x0F	0x8F	Fifo threshold, TX start condition
0x3D	RegPacketConfig2	0x02		Packet mode settings
0x3E-0x4D	RegAesKey1-16	0x00		16 bytes of the cypher key
0x4E	RegTemp1	0x01		Temperature Sensor control
0x4F	RegTemp2	0x00		Temperature readout
0x58	RegTestLna	0x1B		Sensitivity boost
0x5F	PDSTST3	0x08		Test register includes PLL bandwidth
0x6F	RegTestDagc	0x00	0x30	Fading margin improvement
0x71	RegTestAfc	0x00		AFC offset for low modulation index AFC
0x50 +	RegTest	-		Internal test registers

NOTE

Reset values are automatically refreshed in the chip at Power On Reset.

Default values are the recommended register values, optimizing the device operation.

Registers for which the Default value differs from the Reset value are denoted by a * in the appropriate tables.

7.7 Common Configuration Registers

Table 7-5. Common Configuration Registers (Sheet 1 of 4)

Name (Address)	Bits	Variable Name	Mode	Default Value	Description
RegFifo (0x00)	7-0	Fifo	rw	0x00	FIFO data input/output
RegOpMode (0x01)	7	SequencerOff	rw	0	Controls the automatic Sequencer: 0 → Operating mode as selected with Mode bits in RegOpMode is automatically reached with the Sequencer 1 → Mode is forced by the user
	6	ListenOn	rw	0	Enables Listen mode: 0 → Off 1 → On
	5	ListenAbort	w	0	Aborts Listen mode when set together with ListenOn=0 and new Mode selection in 1 SPI access Always reads 0.
	4-2	Mode	rw	001	Transceiver’s operating modes: 000 → Sleep mode (SLEEP) 001 → Standby mode (STDBY) 010 → Frequency Synthesizer mode (FS) 011 → Transmitter mode (TX) 100 → Receiver mode (RX) others → reserved Reads the value corresponding to the current chip mode
	1-0	-	r	00	unused

Table 7-5. Common Configuration Registers (Sheet 2 of 4)

Name (Address)	Bits	Variable Name	Mode	Default Value	Description
RegDataModul (0x02)	7	-	r	0	unused
	6-5	DataMode	rw	00	Data processing mode: 00 → Packet mode 01 → reserved 10 → Continuous mode with bit synchronizer 11 → Continuous mode without bit synchronizer
	4-3	ModulationType	rw	00	Modulation scheme: 00 → FSK 01 → OOK 10 - 11 → reserved
	2	-	r	0	unused
	1-0	ModulationShaping	rw	00	Data shaping: in FSK: 00 → no shaping 01 → Gaussian filter, BT = 1.0 10 → Gaussian filter, BT = 0.5 11 → Gaussian filter, BT = 0.3 in OOK: 00 → no shaping 01 → filtering with f_{cutoff} = BR 10 → filtering with f_{cutoff} = 2*BR 11 → reserved
RegBitrateMsb (0x03)	7-0	BitRate(15:8)	rw	0x1a	MSB of Bit Rate (Chip Rate when Manchester encoding is enabled)
RegBitrateLsb (0x04)	7-0	BitRate(7:0)	rw	0x0b	LSB of Bit Rate (Chip Rate if Manchester encoding is enabled) $BitRate = \frac{FXOSC}{BitRate(15,0)}$
RegFdevMsb (0x05)	7-6	-	r	00	unused
	5-0	Fdev(13:8)	rw	000000	MSB of the frequency deviation
RegFdevLsb (0x06)	7-0	Fdev(7:0)	rw	0x52	LSB of the frequency deviation $Fdev = Fstep \times Fdev(15,0)$
RegFrfMsb (0x07)	7-0	Frf(23:16)	rw	0xe4	MSB of the RF carrier frequency
RegFrfMid (0x08)	7-0	Frf(15:8)	rw	0xc0	Middle byte of the RF carrier frequency
RegFrfLsb (0x09)	7-0	Frf(7:0)	rw	0x00	LSB of the RF carrier frequency $Frf = Fstep \times Frf(23;0)$

Table 7-5. Common Configuration Registers (Sheet 3 of 4)

Name (Address)	Bits	Variable Name	Mode	Default Value	Description
RegOsc1 (0x0A)	7	RcCalStart	w	0	Triggers the calibration of the RC oscillator when set. Always reads 0. RC calibration must be triggered in Standby mode.
	6	RcCalDone	r	1	0 → RC calibration in progress 1 → RC calibration is over
	5-0	-	r	000001	unused
RegAfcCtrl (0x0B)	7-6	-	r	00	unused
	5	AfcLowBetaOn	rw	0	Improved AFC routine for signals with modulation index lower than 2. 0 → Standard AFC routine 1 → Improved AFC routine
	4-0	-	r	00000	unused
RegLowBat (0x0C)	7-5	-	r	000	unused
	4	LowBatMonitor	rw	-	Real-time (not latched) output of the Low Battery detector, when enabled.
	3	LowBatOn	rw	0	Low Battery detector enable signal 0 → LowBat off 1 → LowBat on
	2-0	LowBatTrim	rw	010	Trimming of the LowBat threshold: 000 → 1.695 V 001 → 1.764 V 010 → 1.835 V 011 → 1.905 V 100 → 1.976 V 101 → 2.045 V 110 → 2.116 V 111 → 2.185 V

Table 7-5. Common Configuration Registers (Sheet 4 of 4)

Name (Address)	Bits	Variable Name	Mode	Default Value	Description
RegListen1 (0x0D)	7-6	ListenResolIdle	rw	10	Resolution of Listen mode Idle time (calibrated RC osc): 00 → reserved 01 → 64 us 10 → 4.1 ms 11 → 262 ms
	5-4	ListenResolRx	rw	01	Resolution of Listen mode RX time (calibrated RC osc): 00 → reserved 01 → 64 us 10 → 4.1 ms 11 → 262 ms
	3	ListenCriteria	rw	0	Criteria for packet acceptance in Listen mode: 0 → signal strength is above *RssiThreshold* 1 → signal strength is above *RssiThreshold* and *SyncAddress* matched
	2-1	ListenEnd	rw	01	Action taken after acceptance of a packet in Listen mode: 00 → chip stays in RX mode. Listen mode stops and must be disabled. 01 → chip stays in RX mode until *PayloadReady* or *Timeout* interrupt occurs. It then goes to the mode defined by *Mode*. Listen mode stops and must be disabled. 10 → chip stays in RX mode until *PayloadReady* or *Timeout* interrupt occurs. Listen mode then resumes in Idle state. FIFO content is lost at next RX wakeup. 11 → Reserved
	0	-	r	0	unused
RegListen2 (0x0E)	7-0	ListenCoefIdle	rw	0xf5	Duration of the Idle phase in Listen mode.
RegListen3 (0x0F)	7-0	ListenCoefRx	rw	0x20	Duration of the RX phase in Listen mode (startup time included
RegVersion (0x10)	7-0	Version	r	0x23	Version code of the chip. Bits 7-4 give the full revision number; bits 3-0 give the metal mask revision number.

7.8 Transmitter Registers

Table 7-6. Transmitter Registers

Name (Address)	Bits	Variable Name	Mode	Default Value	Description
RegPaLevel (0x11)	7	Pa0On *	rw	1	Enables PA0, connected to RFIO and LNA
	6	Pa1On *	rw	0	Enables PA1, on PA_BOOST pin
	5	Pa2On *	rw	0	Enables PA2, on PA_BOOST pin
	4-0	OutputPower	rw	11111	Output power setting, with 1 dB steps Pout = -18 + *OutputPower* [dBm] , with PA0 or PA1 Pout = +2 to +17 dBm, with PA1 and PA2 valid range of OutputPower is 10000 through 11111 only. Do not set to 0xxxx when using PA2.
RegPaRamp (0x12)	7-4	-	r	0000	unused
	3-0	PaRamp	rw	1001	Rise/Fall time of ramp up/down in FSK 0000 → 3.4 ms 0001 → 2 ms 0010 → 1 ms 0011 → 500 us 0100 → 250 us 0101 → 125 us 0110 → 100 us 0111 → 62 us 1000 → 50 us 1001 → 40 us 1010 → 31 us 1011 → 25 us 1100 → 20 us 1101 → 15 us 1110 → 12 us 1111 → 10 us
RegOcp (0x13)	7-5	-	r	000	unused
	4	OcpOn	rw	1	Enables overload current protection (OCP) for the PA: 0 → OCP disabled 1 → OCP enabled
	3-0	OcpTrim	rw	1010	Trimming of OCP current: $Imax = 45 + 5 \times OcpTrim(mA)$ 95 mA OCP by default

NOTE

Power Amplifier truth table is available in Table 5-2.

7.9 Receiver Registers

Table 7-7. Receiver Registers

Name (Address)	Bits	Variable Name	Mode	Default Value	Description
Reserved14 (0x14)	7-0	-	r	0x40	unused
Reserved15 (0x15)	7-0	-	r	0xB0	unused
Reserved16 (0x16)	7-0	-	r	0x7B	unused
Reserved17 (0x17)	7-0	-	r	0x9B	unused
RegLna (0x18)	7	LnaZin	rw	1 *	LNA's input impedance 0 → 50 ohms 1 → 200 ohms
	6	-	r	0	unused
	5-3	LnaCurrentGain	r	001	Current LNA gain, set either manually, or by the AGC
	2-0	LnaGainSelect	rw	000	LNA gain setting: 000 → gain set by the internal AGC loop 001 → G1 = highest gain 010 → G2 = highest gain – 6 dB 011 → G3 = highest gain – 12 dB 100 → G4 = highest gain – 24 dB 101 → G5 = highest gain – 36 dB 110 → G6 = highest gain – 48 dB 111 → reserved
RegRxBw (0x19)	7-5	DccFreq	rw	010 *	Cut-off frequency of the DC offset canceller (DCC): $fc = \frac{4 \times RxBw}{2\pi \times 2^{DccFreq + 2}}$ ~4% of the RxBw by default
	4-3	RxBwMant	rw	10 *	Channel filter bandwidth control: 00 → RxBwMant = 16 10 → RxBwMant = 24 01 → RxBwMant = 20 11 → reserved
	2-0	RxBwExp	rw	101 *	Channel filter bandwidth control: FSK Mode: $RxBw = \frac{FXOSC}{RxBwMant \times 2^{RxBwExp + 2}}$ OOK Mode: $RxBw = \frac{FXOSC}{RxBwMant \times 2^{RxBwExp + 3}}$

Table 7-7. Receiver Registers

RegAfcBw (0x1A)	7-5	DccFreqAfc	rw	100	DccFreq parameter used during the AFC
	4-3	RxBwMantAfc	rw	01	RxBwMant parameter used during the AFC
	2-0	RxBwExpAfc	rw	011 *	RxBwExp parameter used during the AFC
RegOokPeak (0x1B)	7-6	OokThreshType	rw	01	Selects type of threshold in the OOK data slicer: 00 → fixed 10 → average 01 → peak 11 → reserved
	5-3	OokPeakTheshStep	rw	000	Size of each decrement of the RSSI threshold in the OOK demodulator: 000 → 0.5 dB 001 → 1.0 dB 010 → 1.5 dB 011 → 2.0 dB 100 → 3.0 dB 101 → 4.0 dB 110 → 5.0 dB 111 → 6.0 dB
	2-0	OokPeakThreshDec	rw	000	Period of decrement of the RSSI threshold in the OOK demodulator: 000 → once per chip 001 → once every 2 chips 010 → once every 4 chips 011 → once every 8 chips 100 → twice in each chip 101 → 4 times in each chip 110 → 8 times in each chip 111 → 16 times in each chip
RegOokAvg (0x1C)	7-6	OokAverageThreshFilt	rw	10	Filter coefficients in average mode of the OOK demodulator: 00 → f_C ? chip rate / 32.? 01 → f_C ? chip rate / 8.? 10 → f_C ? chip rate / 4.? 11 → f_C ? chip rate / 2.?
	5-0	-	r	000000	unused
RegOokFix (0x1D)	7-0	OokFixedThresh	rw	0110 (6dB)	Fixed threshold value (in dB) in the OOK demodulator. Used when *OokThresType* = 00

Table 7-7. Receiver Registers

RegAfcFei (0x1E)	7	-	r	0	unused
	6	FeiDone	r	0	0 → FEI is on-going 1 → FEI finished
	5	FeiStart	w	0	Triggers a FEI measurement when set. Always reads 0.
	4	AfcDone	r	1	0 → AFC is on-going 1 → AFC has finished
	3	AfcAutoclearOn	rw	0	Only valid if *AfcAutoOn* is set 0 → AFC register is not cleared before a new AFC phase 1 → AFC register is cleared before a new AFC phase
	2	AfcAutoOn	rw	0	0 → AFC is performed each time *AfcStart* is set 1 → AFC is performed each time RX mode is entered
	1	AfcClear	w	0	Clears the AfcValue if set in RX mode. Always reads 0
	0	AfcStart	w	0	Triggers an AFC when set. Always reads 0.
RegAfcMsb (0x1F)	7-0	AfcValue(15:8)	r	0x00	MSB of the AfcValue, 2's complement format
RegAfcLsb (0x20)	7-0	AfcValue(7:0)	r	0x00	LSB of the AfcValue, 2's complement format *Frequency correction* = AfcValue x Fstep
RegFeiMsb (0x21)	7-0	FeiValue(15:8)	r	-	MSB of the measured frequency offset, 2's complement
RegFeiLsb (0x22)	7-0	FeiValue(7:0)	r	-	LSB of the measured frequency offset, 2's complement *Frequency error* = FeiValue x Fstep
RegRssiConfig (0x23)	7-2	-	r	000000	unused
	1	RssiDone	r	1	0 → RSSI is on-going 1 → RSSI sampling is finished, result available
	0	RssiStart	w	0	Trigger a RSSI measurement when set. Always reads 0.
RegRssiValue (0x24)	7-0	RssiValue	r	0xFF	Absolute value of the RSSI in dBm, 0.5dB steps. RSSI = *-RssiValue/2 [dBm]*

7.10 IRQ and Pin Mapping Registers

Table 7-8. IRQ and Pin Mapping Registers

	Bits		**Mode**	**Default Value**	
RegDioMapping1 (0x25)	7-6	Dio0Mapping	rw	00	Mapping of pins DIO0 to DIO5 See Table 7-2 for mapping in Continuous mode See Table 7-3 for mapping in Packet mode
	5-4	Dio1Mapping	rw	00	
	3-2	Dio2Mapping	rw	00	
	1-0	Dio3Mapping	rw	00	
RegDioMapping2 (0x26)	7-6	Dio4Mapping	rw	00	
	5-4	Dio5Mapping	rw	00	
	3	-	r	0	unused
	2-0	ClkOut	rw	111 *	Selects CLKOUT frequency: 000 → FXOSC 001 → FXOSC / 2 010 → FXOSC / 4 011 → FXOSC / 8 100 → FXOSC / 16 101 → FXOSC / 32 110 → RC (automatically enabled) 111 → OFF

Table 7-8. IRQ and Pin Mapping Registers

RegIrqFlags1 (0x27)	7	ModeReady	r	1	Set when the operation mode requested in *Mode*, is ready - Sleep: Entering Sleep mode - Standby: XO is running - FS: PLL is locked - RX: RSSI sampling starts - TX: PA ramp-up completed Cleared when changing operating mode.
	6	RxReady	r	0	Set in RX mode, after RSSI, AGC and AFC. Cleared when leaving RX.
	5	TxReady	r	0	Set in TX mode, after PA ramp-up. Cleared when leaving TX.
	4	PllLock	r	0	Set (in FS, RX or TX) when the PLL is locked. Cleared when it is not.
	3	Rssi	rwc	0	Set in RX when the *RssiValue* exceeds *RssiThreshold.* Cleared when leaving RX.
	2	Timeout	r	0	Set when a timeout occurs (see *TimeoutRxStart* and *TimeoutRssiThresh*) Cleared when leaving RX or FIFO is emptied.
	1	AutoMode	r	0	Set when entering Intermediate mode. Cleared when exiting Intermediate mode. Please note that in Sleep mode a small delay can be observed between *AutoMode* interrupt and the corresponding enter/exit condition.
	0	SyncAddressMatch	r/rwc	0	Set when Sync and Address (if enabled) are detected. Cleared when leaving RX or FIFO is emptied. This bit is read only in Packet mode, rwc in Continuous mode

Table 7-8. IRQ and Pin Mapping Registers

RegIrqFlags2 (0x28)	7	FifoFull	r	0	Set when FIFO is full (i.e. contains 66 bytes), else cleared.
	6	FifoNotEmpty	r	0	Set when FIFO contains at least one byte, else cleared
	5	FifoLevel	r	0	Set when the number of bytes in the FIFO strictly exceeds *FifoThreshold*, else cleared.
	4	FifoOverrun	rwc	0	Set when FIFO overrun occurs. (except in Sleep mode) Flag(s) and FIFO are cleared when this bit is set. The FIFO then becomes immediately available for the next transmission / reception.
	3	PacketSent	r	0	Set in TX when the complete packet has been sent. Cleared when exiting TX.
	2	PayloadReady	r	0	Set in RX when the payload is ready (i.e. last byte received and CRC, if enabled and *CrcAutoClearOff* is cleared, is Ok). Cleared when FIFO is empty.
	1	CrcOk	r	0	Set in RX when the CRC of the payload is Ok. Cleared when FIFO is empty.
	0	LowBat	rwc	-	Set when the battery voltage drops below the Low Battery threshold. Cleared only when set by the user.
RegRssiThresh (0x29)	7-0	RssiThreshold	rw	0xE4 *	RSSI trigger level for *Rssi* interrupt : - *RssiThreshold* / 2 [dBm]
RegRxTimeout1 (0x2A)	7-0	TimeoutRxStart	rw	0x00	*Timeout* interrupt is generated *TimeoutRxStart**16*T_{bit} after switching to RX mode if *Rssi* interrupt doesn't occur (i.e. *RssiValue* > *RssiThreshold)* 0x00: *TimeoutRxStart* is disabled
RegRxTimeout2 (0x2B)	7-0	TimeoutRssiThresh	rw	0x00	*Timeout* interrupt is generated *TimeoutRssiThresh**16*T_{bit} after *Rssi* interrupt if *PayloadReady* interrupt doesn't occur. 0x00: *TimeoutRssiThresh* is disabled

7.11 Packet Engine Registers

Table 7-9. Packet Engine Registers

	Bits		**Mode**	**Default Value**	
RegPreambleMsb (0x2c)	7-0	PreambleSize(15:8)	rw	0x00	Size of the preamble to be sent (from *TxStartCondition* fulfilled). (MSB byte)
RegPreambleLsb (0x2d)	7-0	PreambleSize(7:0)	rw	0x03	Size of the preamble to be sent (from *TxStartCondition* fulfilled). (LSB byte)
RegSyncConfig (0x2e)	7	SyncOn	rw	1	Enables the Sync word generation and detection: 0 → Off 1 → On
	6	FifoFillCondition	rw	0	FIFO filling condition: 0 → if *SyncAddress* interrupt occurs 1 → as long as *FifoFillCondition* is set
	5-3	SyncSize	rw	011	Size of the Sync word: (*SyncSize* + 1) bytes
	2-0	SyncTol	rw	000	Number of tolerated bit errors in Sync word
RegSyncValue1 (0x2f)	7-0	SyncValue(63:56)	rw	0x01 *	1st byte of Sync word. (MSB byte) Used if *SyncOn* is set.
RegSyncValue2 (0x30)	7-0	SyncValue(55:48)	rw	0x01 *	2nd byte of Sync word Used if *SyncOn* is set and *(SyncSize +1)* >= 2.
RegSyncValue3 (0x31)	7-0	SyncValue(47:40)	rw	0x01 *	3rd byte of Sync word. Used if *SyncOn* is set and *(SyncSize +1)* >= 3.
RegSyncValue4 (0x32)	7-0	SyncValue(39:32)	rw	0x01 *	4th byte of Sync word. Used if *SyncOn* is set and *(SyncSize +1)* >= 4.
RegSyncValue5 (0x33)	7-0	SyncValue(31:24)	rw	0x01 *	5th byte of Sync word. Used if *SyncOn* is set and *(SyncSize +1)* >= 5.
RegSyncValue6 (0x34)	7-0	SyncValue(23:16)	rw	0x01 *	6th byte of Sync word. Used if *SyncOn* is set and *(SyncSize +1)* >= 6.
RegSyncValue7 (0x35)	7-0	SyncValue(15:8)	rw	0x01 *	7th byte of Sync word. Used if *SyncOn* is set and *(SyncSize +1)* >= 7.
RegSyncValue8 (0x36)	7-0	SyncValue(7:0)	rw	0x01 *	8th byte of Sync word. Used if *SyncOn* is set and *(SyncSize +1)* = 8.

Table 7-9. Packet Engine Registers

RegPacketConfig1 (0x37)	7	PacketFormat	rw	0	Defines the packet format used: 0 → Fixed length 1 → Variable length
	6-5	DcFree	rw	00	Defines DC-free encoding/decoding performed: 00 → None (Off) 01 → Manchester 10 → Whitening 11 → reserved
	4	CrcOn	rw	1	Enables CRC calculation/check (TX/RX): 0 → Off 1 → On
	3	CrcAutoClearOff	rw	0	Defines the behavior of the packet handler when CRC check fails: 0 → Clear FIFO and restart new packet reception. No *PayloadReady* interrupt issued. 1 → Do not clear FIFO. *PayloadReady* interrupt issued.
	2-1	AddressFiltering	rw	00	Defines address based filtering in RX: 00 → None (Off) 01 → Address field must match *NodeAddress* 10 → Address field must match *NodeAddress* or *BroadcastAddress* 11 → reserved
	0	-	rw	0	unused
RegPayloadLength (0x38)	7-0	PayloadLength	rw	0x40	If PacketFormat = 0 (fixed), payload length. If PacketFormat = 1 (variable), max length in RX, not used in TX.
RegNodeAdrs (0x39)	7-0	NodeAddress	rw	0x00	Node address used in address filtering.
RegBroadcastAdrs (0x3A)	7-0	BroadcastAddress	rw	0x00	Broadcast address used in address filtering.

Table 7-9. Packet Engine Registers

RegAutoModes (0x3B)	7-5	EnterCondition	rw	000	Interrupt condition for entering the intermediate mode: 000 → None (AutoModes Off) 001 → Rising edge of *FifoNotEmpty* 010 → Rising edge of *FifoLevel* 011 → Rising edge of *CrcOk* 100 → Rising edge of *PayloadReady* 101 → Rising edge of *SyncAddress* 110 → Rising edge of *PacketSent* 111 → Falling edge of *FifoNotEmpty* (i.e. FIFO empty)
	4-2	ExitCondition	rw	000	Interrupt condition for exiting the intermediate mode: 000 → None (AutoModes Off) 001 → Falling edge of *FifoNotEmpty* (i.e. FIFO empty) 010 → Rising edge of *FifoLevel* or *Timeout* 011 → Rising edge of *CrcOk* or *Timeout* 100 → Rising edge of *PayloadReady* or *Timeout* 101 → Rising edge of *SyncAddress* or *Timeout* 110 → Rising edge of *PacketSent* 111 → Rising edge of *Timeout*
	1-0	IntermediateMode	rw	00	Intermediate mode: 00 → Sleep mode (SLEEP) 01 → Standby mode (STDBY) 10 → Receiver mode (RX) 11 → Transmitter mode (TX)
RegFifoThresh (0x3C)	7	TxStartCondition	rw	1 *	Defines the condition to start packet transmission : 0 → *FifoLevel* (i.e. the number of bytes in the FIFO exceeds *FifoThreshold)* 1 → *FifoNotEmpty* (i.e. at least one byte in the FIFO)
	6-0	FifoThreshold	rw	000111 1	Used to trigger *FifoLevel* interrupt.
RegPacketConfig2 (0x3D)	7-4	InterPacketRxDelay	rw	0000	After *PayloadReady* occurred, defines the delay between FIFO empty and the start of a new RSSI phase for next packet. Must match the transmitter's PA ramp-down time. - Tdelay = 0 if *InterpacketRxDelay* >= 12 - Tdelay = ($2^{InterpacketRxDelay}$) / *BitRate* otherwise
	3	-	rw	0	unused
	2	RestartRx	w	0	Forces the Receiver in WAIT mode, in Continuous RX mode. Always reads 0.
	1	AutoRxRestartOn	rw	1	Enables automatic RX restart (RSSI phase) after *PayloadReady* occurred and packet has been completely read from FIFO: 0 → Off. *RestartRx* can be used. 1 → On. RX automatically restarted after *InterPacketRxDelay*.
	0	AesOn	rw	0	Enable the AES encryption/decryption: 0 → Off 1 → On (payload limited to 66 bytes maximum)
RegAesKey1 (0x3E)	7-0	AesKey(127:120)	w	0x00	1st byte of cipher key (MSB byte)

Table 7-9. Packet Engine Registers

RegAesKey2 (0x3F)	7-0	AesKey(119:112)	w	0x00	2nd byte of cipher key
RegAesKey3 (0x40)	7-0	AesKey(111:104)	w	0x00	3rd byte of cipher key
RegAesKey4 (0x41)	7-0	AesKey(103:96)	w	0x00	4th byte of cipher key
RegAesKey5 (0x42)	7-0	AesKey(95:88)	w	0x00	5th byte of cipher key
RegAesKey6 (0x43)	7-0	AesKey(87:80)	w	0x00	6th byte of cipher key
RegAesKey7 (0x44)	7-0	AesKey(79:72)	w	0x00	7th byte of cipher key
RegAesKey8 (0x45)	7-0	AesKey(71:64)	w	0x00	8th byte of cipher key
RegAesKey9 (0x46)	7-0	AesKey(63:56)	w	0x00	9th byte of cipher key
RegAesKey10 (0x47)	7-0	AesKey(55:48)	w	0x00	10th byte of cipher key
RegAesKey11 (0x48)	7-0	AesKey(47:40)	w	0x00	11th byte of cipher key
RegAesKey12 (0x49)	7-0	AesKey(39:32)	w	0x00	12th byte of cipher key
RegAesKey13 (0x4A)	7-0	AesKey(31:24)	w	0x00	13th byte of cipher key
RegAesKey14 (0x4B)	7-0	AesKey(23:16)	w	0x00	14th byte of cipher key
RegAesKey15 (0x4C)	7-0	AesKey(15:8)	w	0x00	15th byte of cipher key
RegAesKey16 (0x4D)	7-0	AesKey(7:0)	w	0x00	16th byte of cipher key (LSB byte)

7.12 Temperature Sensor Registers

Table 7-10. Temperature Sensor Registers

Name (Address)	Bits	Variable Name	Mode	Default Value	Description
RegTemp1 (0x4E)	7-4	-	r	0000	unused
	3	TempMeasStart	w	0	Triggers the temperature measurement when set. Always reads 0.
	2	TempMeasRunning	r	0	Set to 1 while the temperature measurement is running. Toggles back to 0 when the measurement has completed. The receiver can not be used while measuring temperature
	1-0	-	r	01	unused
RegTemp2 (0x4F)	7-0	TempValue	r	-	Measured temperature -1°C per Lsb Needs calibration for accuracy

7.13 Test Registers

Table 7-11. Test Registers

Name (Address)	Bits	Variable Name	Mode	Default Value	Description
RegTestLna (0x58)	7-0	SensitivityBoost	rw	0x1B	High sensitivity or normal sensitivity mode: 0x1B → Normal mode 0x2D → High sensitivity mode
RegTestPLL (0x5F)	7-0	PLL Bandwidth	rw	0x0C	PLL Bandwidth Adjust 00 = 75 kHz 04 = 150 kHz 08 = 300 kHz, default (for some regions), for bit rates up to 300 kbps 0C = 600 kHz for bit rates up to 600 kbps
RegTestDagc (0x6F)	7-0	ContinuousDagc	rw	0x30	Fading Margin Improvement, refer to Section 5.7.3, "Continuous-Time DAGC" 0x00 → Normal mode 0x10 → Improved margin, use if Afc*LowBetaAfcOn = 1* 0x30 → Improved margin, use if Afc*LowBetaAfcOn = 0*
RegTestAfc (0x71)	7-0	LowBetaAfcOffset	rw	0x00	AFC offset set for low modulation index systems, used if *AfcLowBetaOn=1*. *Offset = LowBetaAfcOffset x 488 Hz*

Chapter 8
MKW01Z128 Transceiver - MCU SPI Interface

The MKW01 transceiver and CPU communicate primarily through the onboard SPI interface. MCU has SPI0 is dedicated to the transceiver SPI interface and should not be used for other applications. The transceiver is a SPI slave only, and the MCU SPI module must be programmed and used as a master only.

8.1 SiP Level SPI Pin Connections

The SiP level SPI pin connections are all internal to the device. Figure 8-1 shows all the SiP interconnections with the SPI bus highlighted.

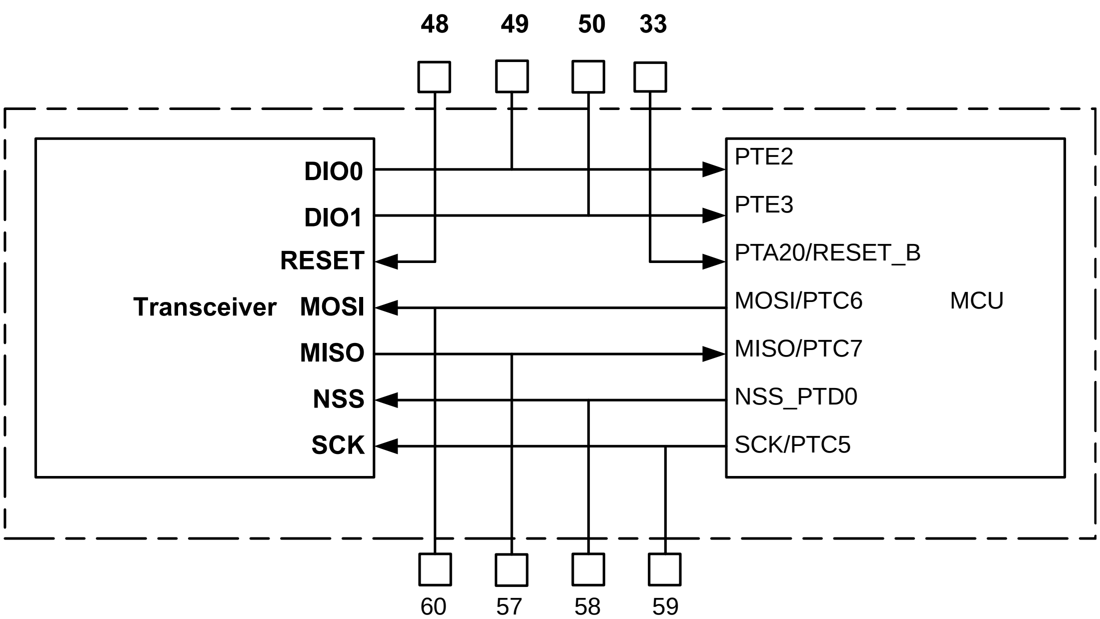


Figure 8-1. MKW01 Internal Interconnects Highlighting SPI Bus

Table 2 in Chapter 2 provides a complete listing of the MKW01 onboard device interconnects.

NOTE

- The MCU SPI port does not default to Port E connections, and these connections MUST be programmed for proper device communication. See Section 2.3, “Internal Functional Interconnects”.
- The SPI external access Pins 57-60 are located on the bottom of the package.

8.2 Features

Features of the SPI bus interface:

- MCU is the SPI Bus master
- Transceiveris bus slave
- Bi-directional data transfer
- Dedicated interface; must meet transceiver protocol requirements
- Programmable SPI clock rate; maximum transfer rate is 10 MHz as determined by the transceiver
- Double-buffered transmit and receive at MCU
- Serial clock phase and polarity must meet transceiver requirements (MCU control bits CPHA = 0 and CPOL = 0)
- Slave select programmed to meet transceiver protocol
- MSB-first shifting

8.3 SPI System Block Diagram

This section shows the system level diagram for the SPI. Figure 8-2 shows the SPI modules of the MCU and transceiver in the master-slave arrangement.

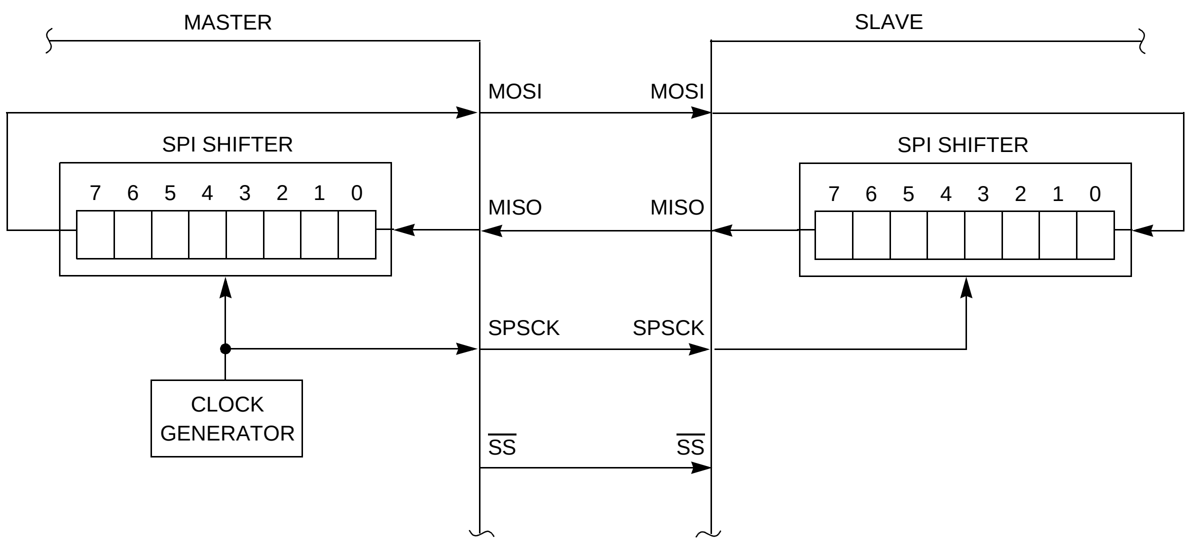


Figure 8-2. SPI System Block Diagram

The MCU (master) initiates all SPI transfers. During a transfer, the master shifts data out (on the MOSI pin) to the slave while simultaneously shifting data in (on the MISO pin) from the slave. Although the SPI interface supports simultaneous data exchange between master and slave, the transceiver SPI protocol only uses data exchange in one direction at a time. The SPSCK signal is a clock output from the master and an input to the slave. The slave device must be selected by a low level on the slave select input ($\overline{SS}$ pin).

8.3.1 SPI Signal Definitions

The SPI signals of $\overline{\text{SS}}$, SCK, MOSI, and MISO are defined in the following paragraphs.

8.3.1.1 Slave Select ($\overline{\text{SS}}$ or NSS)

A transaction on the SPI port is framed by the active low Slave Select ($\overline{\text{SS}}$) input signal which is driven by the MCU as master.

8.3.1.2 SPI Clock (SCK or SPSCK)

The host drives the SPI Clock (SCK) input to the transceiver. Data is clocked into the master or slave on the leading (rising) edge of the return-to-zero SPSCK and data changes state on the trailing (falling) edge of SPSCK.

NOTE

- The SPI Bus protocol as defined by the Motorola/Freescale standard supports other clock/data timing, however, the described mode is the only one used.
- For the MKW0xxx microcontroller, the SPI clock format is the clock phase control bit CPHA = 0 and the clock polarity control bit CPOL = 0.

8.3.1.3 Master Out / Slave In (MOSI)

The Master Out/Slave In (MOSI) signal presents incoming data from the host to the transceiver (slave input).

8.3.1.4 Master In / Slave Out (MISO)

The Master In/Slave Out (MISO) signal presents incoming data from the transceiver to the MCU (master input).

8.3.2 MKW0xxx SPI Transaction Protocol

Although standard SPI protocol is based on 8-bit transfers, the transceiver imposes a higher level transaction protocol that is based on multiple 8-bit transfers per transaction. There are three SPI transaction modes defined to access the transceiver registers:

- SINGLE access - an address byte is followed by a data byte
  - For a write access, the data byte (MOSI) is written to the addressed transceiver register
  - For a read access, the data byte (MISO) is read from the addressed transceiver register
  - The NSS pin goes low at the beginning of the frame and goes high after the data byte
- BURST access - an address byte is followed by several data bytes.
  - The address byte provides the starting register address for the data burst
  - The address is automatically incremented internally between each data byte.
  - This mode supports both read and write accesses.

  - The NSS pin goes low at the beginning of the frame and stays low between bytes. It goes high only after the last byte transfer.
- FIFO access - special case of burst access for the FIFO
  - The address byte corresponds to the address of the FIFO
  - The address is not automatically incremented, but stays pointed at the FIFO.
  - The data bytes are sequentially written to or read from the FIFO.
  - The NSS pin goes low at the beginning of the frame and stays low between bytes. It goes high only after the last byte transfer

For the defined transaction formats:

- The falling edge of NSS always initializes the start of the frame transfer.
- All address and data sent MSB first
- The address byte (first byte) of the transaction is composed of -
  - WNR Bit (Bit 7) - which is “1” for write access and “0” for read access
  - Address (Bit 6:0) - 7-bit register address (sent MSB first)
- Following data byte(s) -
  - Write data sent on MOSI
  - Read data sent on MISO
- The rising edge of NSS signifies the end of the frame transfer (NSS must remain asserted for the entire frame).

8.3.3 MKW0xxx SPI Transaction Timing

As defined in Section 8.3.2, the SPI transaction protocol is composed of two or more bytes per frame. Although the transceiver is capable of a continuous bit transfer for the entire frame, the MCU SPI port only transfers data in bursts of 8 bits. There are implications in the way the MCU SPI is programmed and used:

- The MCU SS signal -
  - Cannot be programmed for SPI module master mode driven operation
  - Port signal PTE3 must be enabled and programmed as a GPIO output to provide the required SS signal timing
- The transaction bytes are sent as a bursts as allowed by the MCU 8-bit SPI module.

Because the MCU is embedded in the SiP and the transceiver only supports the one clock format, the MCU SPI must be programmed for this clock mode, i.e., clock phase control bit CPHA = 0 and the clock polarity control bit CPOL = 0. In addition, the MSB-first option must be selected.

Figure 8-3 illustrates a simple single read access transaction timing. The top part of the figure shows the SPI timing for a single byte transfer. For the complete frame, the SS signal goes low and stays low for both byte transfers. The byte transfers are actually accomplished as two operations for the MCI SPI peripheral block.

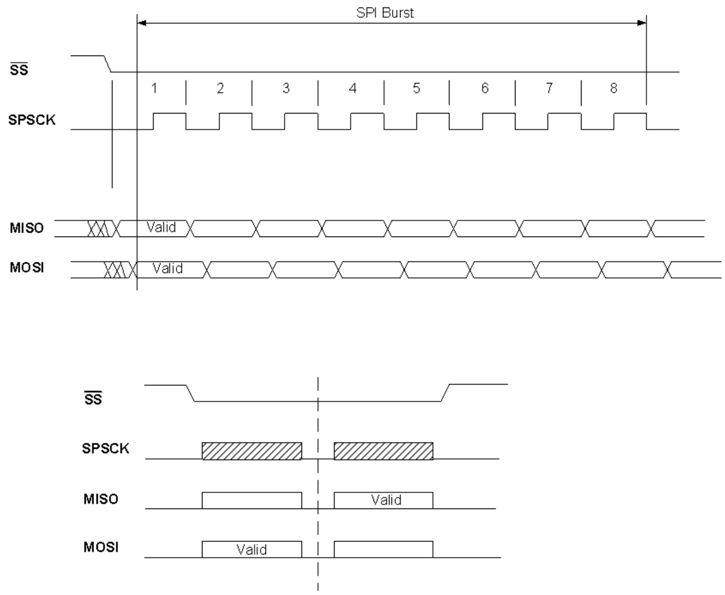


Figure 8-3. Transceiver SPI Read Single Access Timing Diagram

The SPI Bus timing is determined by the transceiver specification as given in Table 8-1.

Table 8-1. Transceiver SPI Timing Specifications

Parameter	Min	Typ	Max	Unit
SPSCK period	100			ns
Pulse width, SPSCK low	50			ns
Pulse width, SPSCK high	50			ns
Setup time, $\overline{\text{SS}}$ low to rising SPSCK	30			ns
Hold time, from falling SPSCK to $\overline{\text{SS}}$ rising edge	60			ns
Setup time, MOSI valid to rising SPSCK	30			ns
Hold time, MOSI valid from rising SPSCK	60			ns
SS high time between accesses	20			ns

NOTE

For the onboard MCU, the bus clock to the SPI module is always ½ the CPU clock, and in turn, the maximum SPI clock baud rate is ½ the peripheral clock. The default bus clock rate for some versions of Freescale supplied software is 32 MHz which in turn provides an 8 MHz SPI baud rate.

Appendix A
MKW01xx MCU Reference Manual

This appendix contains reference material pulled from the reference manual for the MCU in this System in Package (SiP).

MKW01Z128 MCU Reference Manual

Sub 1 GHz Low-Power Transceiver plus Microcontroller Reference Manual - Microcontroller Information

Document Number: MKW01xxRM

Contents

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

Chapter 1
Chip Configuration

Section number	Title	Page

Chapter 2
Memory Map

Section number	Title	Page

Chapter 3
Clock Distribution

Chapter 4
Reset and Boot

Section number	Title	Page

Chapter 5
Power Management

Chapter 6
Debug

Section number	Title	Page

Chapter 9
System Mode Controller (SMC)

Section number	Title	Page

Chapter 10
Power Management Controller (PMC)

Chapter 11
Low-Leakage Wakeup Unit (LLWU)

Section number	Title	Page

Chapter 17
Peripheral Bridge (AIPS-Lite)

Chapter 18
Direct Memory Access Multiplexer (DMAMUX)

Chapter 19
DMA Controller Module

Chapter 20
Multipurpose Clock Generator (MCG)

Section number	Title	Page

Chapter 21
Oscillator (OSC)

Section number	Title	Page

Chapter 22
Flash Memory Controller (FMC)

Chapter 23
Flash Memory Module (FTFA)

Chapter 24
Analog-to-Digital Converter (ADC)

Section number	Title	Page

Chapter 25
Comparator (CMP)

Chapter 26
12-bit Digital-to-Analog Converter (DAC)

Chapter 27
Timer/PWM Module (TPM)

Section number	Title	Page

Chapter 28
Periodic Interrupt Timer (PIT)

Chapter 29
Low-Power Timer (LPTMR)

Section number	Title	Page

Chapter 30
Real Time Clock (RTC)

Section number	Title	Page

Section number	Title	Page

Chapter 32
Universal asynchronous receiver/transmitter (UART)

Section number	Title	Page

Chapter 33
Universal Asynchronous Receiver/Transmitter (UART)

Chapter 34
General-Purpose Input/Output (GPIO)

Chapter 35
Touch Sensing Input (TSI)

Chapter 1
Chip Configuration

1.1 Introduction

Information found here provides details on the individual modules of the microcontroller.

It includes:

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

1.2 Module to module interconnects

1.2.1 Interconnection overview

The following table captures the module to module interconnections for this device.

Table 1-1. Module-to-module interconnects

Peripheral	Signal	—	to Peripheral	Use Case	Control	Comment
TPM1	CH0F, CH1F	to	ADC (Trigger)	ADC Triggering (A AND B)	SIM_SOPT7[ADC0ALTTRGEN] = 0	Ch0 is A, and Ch1 is B, selecting this ADC trigger is for supporting A and B triggering. In Stop and VLPS modes, the second trigger must be set to >10 μs after the first trigger

Table continues on the next page...

Table 1-1. Module-to-module interconnects (continued)

Peripheral	Signal	—	to Peripheral	Use Case	Control	Comment
LPTMR	TCF	to	ADC (Trigger)	ADC Triggering (A or B)	SIM_SOPT7[ADC0TRGSEL] and SIM_SOPT7[ADC0PRETRGSEL] to select A or B	—
TPMx	TOF	to	ADC (Trigger)	ADC Triggering (A or B)	SIM_SOPT7[ADC0TRGSEL], SIM_SOPT7[ADC0PRETRGSEL] to select A or B	—
PIT CHx	TIF0, TIF1	to	ADC (Trigger)	ADC Triggering (A or B)	SIM_SOPT7[ADC0TRGSEL], SIM_SOPT7[ADC0PRETRGSEL] to select A or B	—
RTC	TAF or Seconds	to	ADC (Trigger)	ADC Triggering (A or B)	SIM_SOPT7[ADC0TRGSEL], SIM_SOPT7[ADC0PRETRGSEL] to select A or B	—
EXTRG_IN	EXTRG_IN	to	ADC (Trigger)	ADC Triggering (A or B)	SIM_SOPT7[ADC0TRGSEL], SIM_SOPT7[ADC0PRETRGSEL] to select A or B	—
CMP0	CMP0_OUT	to	ADC (Trigger)	ADC Triggering (Aor B)	SIM_SOPT7[ADC0TRGSEL], SIM_SOPT7[ADC0PRETRGSEL] to select A or B	—
CMP0	CMP0_OUT	to	LPTMR_ALT 0	Count CMP events	LPTMR_CSR[TPS]	—
CMP0	CMP0_OUT	to	TPM1 CH0	Input capture	SIM_SOPT4[TPM1CH0SRC]	—
CMP0	CMP0_OUT	to	TPM2 CH0	Input capture	SIM_SOPT4[TPM2CH0SRC]	—
CMP0	CMP0_OUT	to	UART0_RX	IR interface	SIM_SOPT5[UART0RXSRC]	
CMP0	CMP0_OUT	to	UART1_RX	IR Interface	SIM_SOPT5[UART1RXSRC]	
LPTMR	Hardware trigger	to	CMPx	Low power triggering of the comparator	CMP_CR1[TRIGM]	—
LPTMR	Hardware trigger	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4 bit field)	—
TPMx	TOF	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4-bit field)	—
TPM1	Timebase	to	TPMx	TPM Global timebase input	TPMx_CONF[GTBEEN]	—
PIT CHx	TIF0, TIF1	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4-bit field)	If PIT is triggering the TPM, the TPM clock must be faster than Bus clock.
RTC	ALARM or SECONDS	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4-bit field)	—
EXTRG_IN	EXTRG_IN	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4-bit field)	—
CMP0	CMP0_OUT	to	TPMx	TPM Trigger input	TPMx_CONF[TRGSEL] (4-bit field)	—

Table continues on the next page...

Table 1-1. Module-to-module interconnects (continued)

Peripheral	Signal	—	to Peripheral	Use Case	Control	Comment
LPTMR	Hardware trigger	to	TSI	TSI triggering	TSI selects HW trigger	—
UART0	UART0_TX	to	Modulated by TPM1 CH0	UART modulation	SIM_SOPT5[UART0TXSRC]	
UART0	UART0_TX	to	Modulated by TPM2 CH0	UART modulation	SIM_SOPT5[UART0TXSRC]	
UART1	UART1_TX	to	Modulated by TPM1 CH0	UART modulation	SIM_SOPT5[UART1TXSRC]	
UART1	UART1_TX	to	Modulated by TPM2 CH0	UART modulation	SIM_SOPT5[UART1TXSRC]	
PIT	TIF0	to	DAC	Advance DAC FIFO	DAC HWTRG Select	—
PIT	TIF0	to	DMA CH0	DMA HW Trigger	DMA MUX register option	—
PIT	TIF1	to	DMA CH1	DMA HW Trigger	DMA MUX register option	—

1.2.2 Analog reference options

Several analog blocks have selectable reference voltages as shown in Table 1-2. These options allow analog peripherals to share or have separate analog references. Care should be taken when selecting analog references to avoid cross talk noise.

Table 1-2. Analog reference options

Module	Reference option	Comment/ Reference selection
16-bit SAR ADC	1 - VREFH 2 - VDDA 3 - Reserved	Selected by ADCx_SC2[REFSEL]
12-bit DAC	1 - VREFH 2 - VDDA[1]	Selected by DACx_C0[DACRFS] bit
CMP with 6-bit DAC	Vin1 - VREFH Vin2 - VDD [1]	Selected by CMPx_DACCR[VRSEL]

1. Use this option for the best ADC operation.

1.3 Core modules

1.3.1 ARM Cortex-M0+ core configuration

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

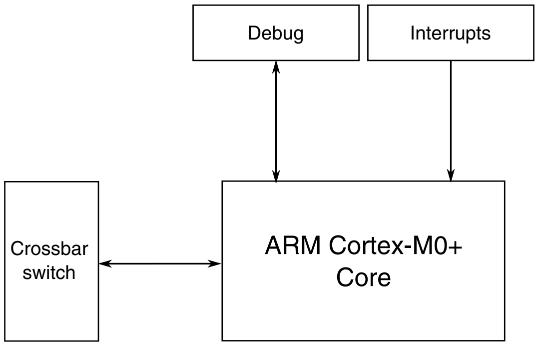


Figure 1-1. Core configuration

Table 1-3. Reference links to related information

Topic	Related module	Reference
Full description	ARM Cortex-M0+ core, r0p0	ARM Cortex-M0+ Technical Reference Manual, r0p0
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
System/instruction/data bus module	Crossbar switch	Crossbar switch
Debug	Serial wire debug (SWD)	Debug
Interrupts	Nested vectored interrupt controller (NVIC)	NVIC
	Miscellaneous control module (MCM)	MCM

1.3.1.1 ARM Cortex M0+ core

The ARM Cortex M0+ parameter settings are as follows:

Table 1-4. ARM Cortex-M0+ parameter settings

Parameter	Verilog name	Value	Description
Arch Clock Gating	ACG	1 = Present	Implements architectural clock gating
DAP Slave Port Support	AHBSLV	1	Supports any AHB debug access port (like the CM4 DAP)

Table continues on the next page...

Table 1-4. ARM Cortex-M0+ parameter settings (continued)

Parameter	Verilog name	Value	Description
DAP ROM Table Base	BASEADDR	0xF000_2003	Base address for DAP ROM table
Endianess	BE	0	Little endian control for data transfers
Breakpoints	BKPT	2	Implements 2 breakpoints
Debug Support	DBG	1 = Present	—
Halt Event Support	HALTEV	1 = Present	—
I/O Port	IOP	1 = Present	Implements single-cycle ld/st accesses to special address space
IRQ Mask Enable	IRQDIS	0x00000000	Assume (for now) all 32 IRQs are used (set if IRQ is disabled)
Debug Port Protocol	JTAGnSW	0 = SWD	SWD protocol, not JTAG
Core Memory Protection	MPU	0 = Absent	No MPU
Number of IRQs	NUMIRQ	32	Assume full NVIC request vector
Reset all registers	RAR	0 = Standard	Do not force all registers to be async reset
Multiplier	SMUL	0 = Fast Mul	Implements single-cycle multiplier
Multi-drop Support	SWMD	0 = Absent	Do not include serial wire support for multi-drop
System Tick Timer	SYST	1 = Present	Implements system tick timer (for CM4 compatibility)
DAP Target ID	TARGETID	0	—
User/Privileged	USER	1 = Present	Implements processor operating modes
Vector Table Offset Register	VTOR	1 = Present	Implements relocation of exception vector table
WIC Support	WIC	1 = Present	Implements WIC interface
WIC Requests	WICLINES	34	Exact number of wake-up IRQs is 34
Watchpoints	WPT	2	Implements two watchpoints

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

1.3.1.2 Buses, interconnects, and interfaces

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

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

1.3.1.3 System tick timer

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

1.3.1.4 Debug facilities

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

1.3.1.5 Core privilege levels

The core on this device is implemented with both privileged and unprivileged levels. The ARM documentation uses different terms than this document to distinguish between privilege levels.

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

1.3.2 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**.

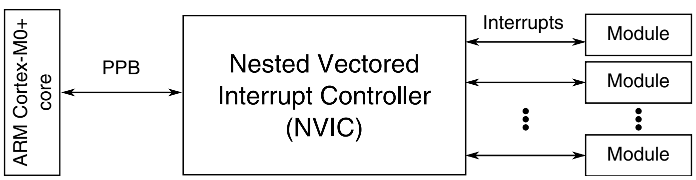


Figure 1-2. NVIC configuration

Table 1-5. Reference links to related information

Topic	Related module	Reference
Full description	Nested vectored interrupt controller (NVIC)	ARM Cortex-M0+ Technical Reference Manual
System memory map	—	System memory map
Clocking	—	Clock distribution

Table continues on the next page...

Table 1-5. Reference links to related information (continued)

Topic	Related module	Reference
Power management	—	Power management
Private peripheral bus (PPB)	ARM Cortex-M0+ core	ARM Cortex-M0+ core

1.3.2.1 Interrupt priority levels

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

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

1.3.2.2 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.

1.3.2.3 Interrupt channel assignments

The interrupt vector assignments are defined in the following table.

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

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

NOTE

The NVIC wake-up sources in the following table support only down to VLPS.

Table 1-7. Interrupt vector assignments

Address	Vector	IRQ[1]	NVIC IPR register number[2]	Source module	Source description
ARM core system handler vectors					
0x0000_0000	0	—	—	ARM core	Initial stack pointer
0x0000_0004	1	—	—	ARM core	Initial program counter
0x0000_0008	2	—	—	ARM core	Non-maskable interrupt (NMI)
0x0000_000C	3	—	—	ARM core	Hard fault
0x0000_0010	4	—	—	—	—
0x0000_0014	5	—	—	—	—
0x0000_0018	6	—	—	—	—
0x0000_001C	7	—	—	—	—
0x0000_0020	8	—	—	—	—
0x0000_0024	9	—	—	—	—
0x0000_0028	10	—	—	—	—
0x0000_002C	11	—	—	ARM core	Supervisor call (SVCall)
0x0000_0030	12	—	—	—	—
0x0000_0034	13	—	—	—	—
0x0000_0038	14	—	—	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	—	—	ARM core	System tick timer (SysTick)
Non-Core Vectors					
0x0000_0040	16	0	0	DMA	DMA channel 0 transfer complete and error
0x0000_0044	17	1	0	DMA	DMA channel 1 transfer complete and error
0x0000_0048	18	2	0	DMA	DMA channel 2 transfer complete and error
0x0000_004C	19	3	0	DMA	DMA channel 3 transfer complete and error
0x0000_0050	20	4	1	—	—
0x0000_0054	21	5	1	FTFA	Command complete and read collision
0x0000_0058	22	6	1	PMC	Low-voltage detect, low-voltage warning
0x0000_005C	23	7	1	LLWU	Low Leakage Wakeup
0x0000_0060	24	8	2	I^2C0	—
0x0000_0064	25	9	2	I^2C1	—
0x0000_0068	26	10	2	SPI0	Single interrupt vector for all sources
0x0000_006C	27	11	2	SPI1	Single interrupt vector for all sources
0x0000_0070	28	12	3	UART0	Status and error
0x0000_0074	29	13	3	UART1	Status and error
0x0000_0078	30	14	3	UART2	Status and error
0x0000_007C	31	15	3	ADC0	—
0x0000_0080	32	16	4	CMP0	—
0x0000_0084	33	17	4	TPM0	—
0x0000_0088	34	18	4	TPM1	—

Table continues on the next page...

Table 1-7. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC IPR register number[2]	Source module	Source description
0x0000_008C	35	19	4	TPM2	—
0x0000_0090	36	20	5	RTC	Alarm interrupt
0x0000_0094	37	21	5	RTC	Seconds interrupt
0x0000_0098	38	22	5	PIT	Single interrupt vector for all channels
0x0000_009C	39	23	5	I^2S0	Single interrupt vector for all sources
0x0000_00A0	40	24	6	—	—
0x0000_00A4	41	25	6	DAC0	—
0x0000_00A8	42	26	6	TSI0	—
0x0000_00AC	43	27	6	MCG	—
0x0000_00B0	44	28	7	LPTMR0	—
0x0000_00B4	45	29	7	—	—
0x0000_00B8	46	30	7	Port control module	Pin detect (Port A)
0x0000_00BC	47	31	7	Port control module	Pin detect (Single interrupt vector for Port C and Port D)

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

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

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

Table 1-8. Interrupt vector assignments

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

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

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

  Since the NVICIPR fields are 2-bit wide (4 priority levels), the NVICIPR2 field range is 22–23.

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

- NVICIPR2[23:22]

1.3.3 Asynchronous wake-up interrupt controller (AWIC) configuration

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

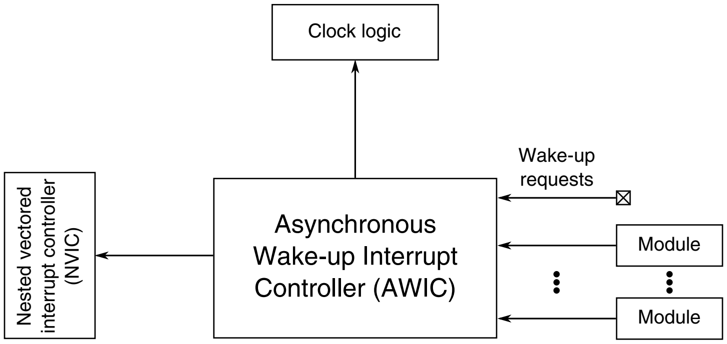


Figure 1-3. Asynchronous wake-up interrupt controller configuration

Table 1-9. 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

1.3.3.1 Wake-up sources

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

Table 1-10. AWIC stop wake-up sources

Wake-up source	Description
Available system resets	$\overline{\text{RESET}}$ pin when LPO is its clock source
Low-voltage detect	Power management controller—functional in Stop mode
Low-voltage warning	Power management controller—functional in Stop mode

Table continues on the next page...

Table 1-10. AWIC stop wake-up sources (continued)

Wake-up source	Description
Pin interrupts	Port control module—any enabled pin interrupt is capable of waking the system.
ADC	The ADC is functional when using internal clock source.
CMP0	Interrupt in normal or trigger mode
I^2Cx	Address match wakeup
UART0	Any interrupt provided clock remains enabled.
UART1 and UART2	Active edge on RXD
RTC	Alarm or seconds interrupt
TSI	Any interrupt
NMI	$\overline{\text{NMI}}$ pin
TPMx	Any interrupt provided clock remains enabled.
LPTMR	Any interrupt provided clock remains enabled.
SPI	Slave mode interrupt

1.4 System modules

1.4.1 SIM configuration

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

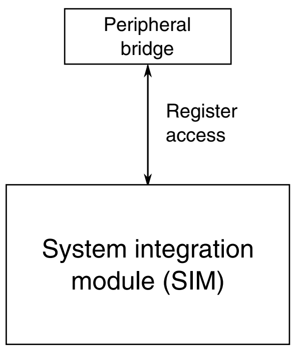


Figure 1-4. SIM configuration

Table 1-11. Reference links to related information

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

1.4.2 System mode controller (SMC) configuration

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

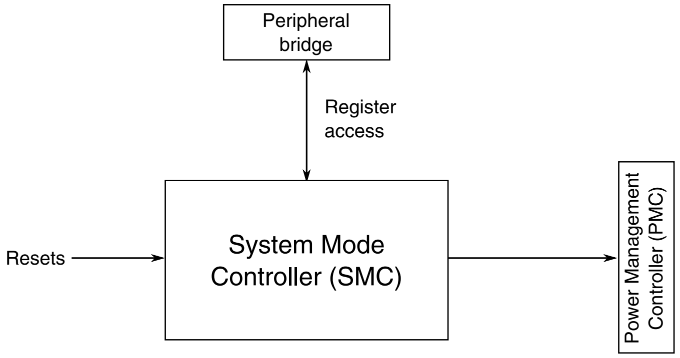


Figure 1-5. System mode controller configuration

Table 1-12. Reference links to related information

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

1.4.2.1 VLLS2 not supported

VLLS2 power mode is not supported on this device.

1.4.3 PMC configuration

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

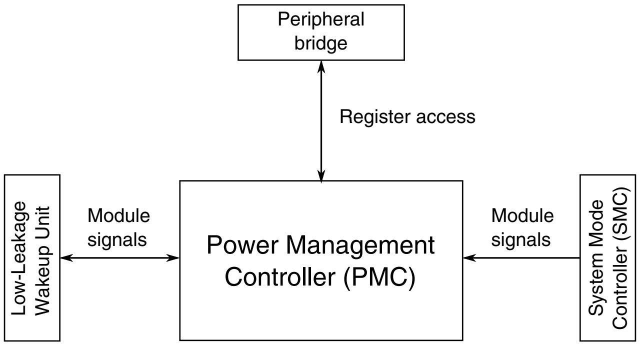


Figure 1-6. PMC configuration

Table 1-13. Reference links to related information

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

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

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

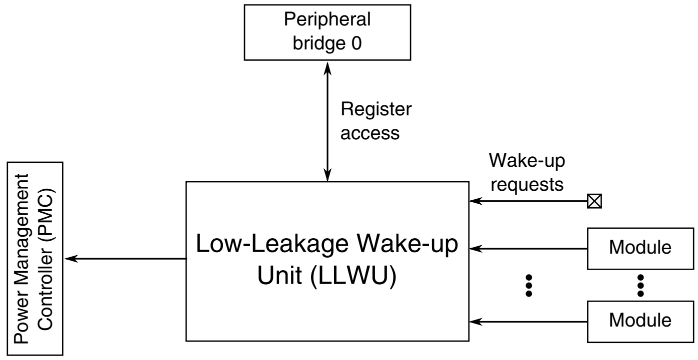


Figure 1-7. Low-Leakage Wake-up Unit configuration

Table 1-14. Reference links to related information

Topic	Related module	Reference
Full description	LLWU	LLWU
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management chapter
	Power Management Controller (PMC)	Power Management Controller (PMC)
	System Mode Controller (SMC)	System Mode Controller
Wake-up requests		LLWU wake-up sources

1.4.4.1 LLWU interrupt

NOTE

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

1.4.4.2 Wake-up Sources

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

NOTE

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

Table 1-15. Wakeup Sources

LLWU pin	Module source or pin name
LLWU_P5	PTB0
LLWU_P6	PTC1
LLWU_P7	PTC3
LLWU_P8	PTC4
Reserved	Reserved
Reserved	Reserved
LLWU_P14	PTD4
LLWU_P15	PTD6
LLWU_M0IF	LPTMR0
LLWU_M1IF	CMP0
LLWU_M2IF	Reserved
LLWU_M3IF	Reserved
LLWU_M4IF	TSI0
LLWU_M5IF	RTC Alarm
LLWU_M6IF	Reserved
LLWU_M7IF	RTC Seconds

1.4.5 MCM configuration

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

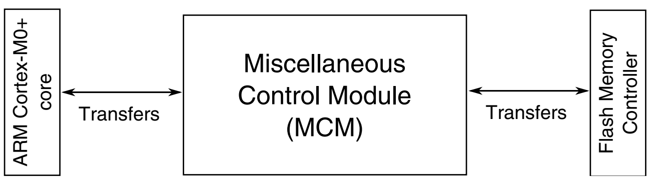


Figure 1-8. MCM configuration

Table 1-16. Reference links to related information

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

Table continues on the next page...

Table 1-16. Reference links to related information (continued)

Topic	Related module	Reference
Transfer	Flash memory controller	Flash memory controller

1.4.6 Crossbar-light switch configuration

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

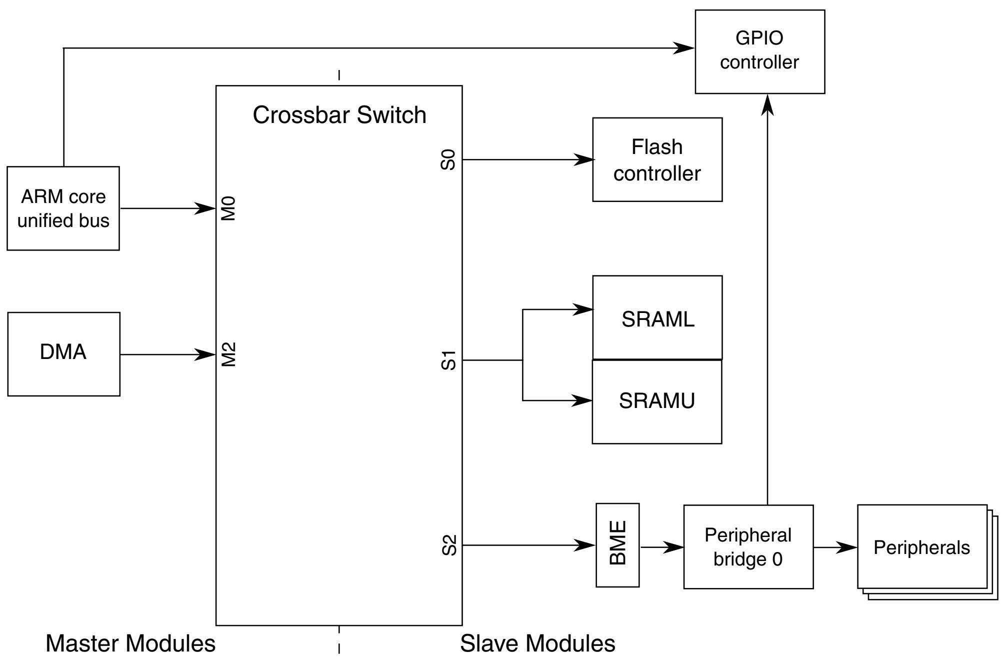


Figure 1-9. Crossbar-light switch integration

Table 1-17. Reference links to related information

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

1.4.6.1 Crossbar-light switch master assignments

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

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

1.4.6.2 Crossbar switch slave assignments

This device contains 3 slaves connected to the crossbar switch.

The slave assignment is as follows:

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

1.4.7 Peripheral bridge configuration

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

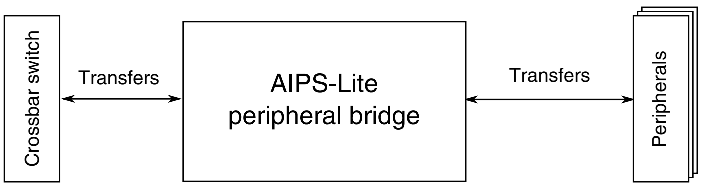


Figure 1-10. Peripheral bridge configuration

Table 1-18. Reference links to related information

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

1.4.7.1 Number of peripheral bridges

This device contains one peripheral bridge.

1.4.7.2 Memory maps

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

1.4.8 DMA request multiplexer configuration

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

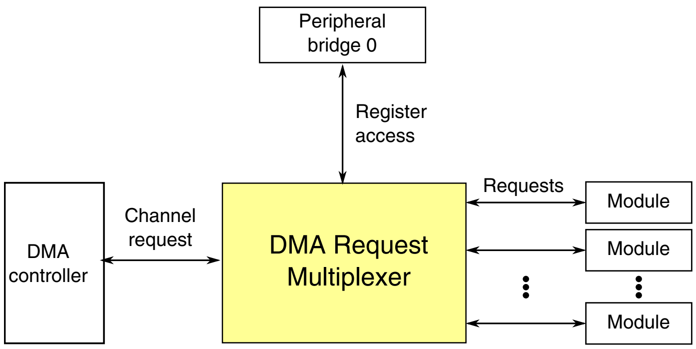


Figure 1-11. DMA request multiplexer configuration

Table 1-19. Reference links to related information

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

1.4.8.1 DMA MUX Request Sources

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

Table 1-20. DMA request sources - MUX 0

Source number	Source module	Source description	Async DMA capable
0	—	Channel disabled[1]	
1	Reserved	Not used	
2	UART0	Receive	Yes
3	UART0	Transmit	Yes
4	UART1	Receive	
5	UART1	Transmit	
6	UART2	Receive	
7	UART2	Transmit	
8	Reserved	—	
9	Reserved	—	
10	Reserved	—	
11	Reserved	—	
12	Reserved	—	
13	Reserved	—	
14	I^2S0	Receive	Yes
15	I^2S0	Transmit	Yes
16	SPI0	Receive	
17	SPI0	Transmit	
18	SPI1	Receive	
19	SPI1	Transmit	
20	Reserved	—	
21	Reserved	—	
22	I^2C0	—	
23	I^2C1	—	
24	TPM0	Channel 0	Yes
25	TPM0	Channel 1	Yes
26	TPM0	Channel 2	Yes
27	TPM0	Channel 3	Yes
28	TPM0	Channel 4	Yes
29	TPM0	Channel 5	Yes
30	Reserved	—	

Table continues on the next page...

Table 1-20. DMA request sources - MUX 0 (continued)

Source number	Source module	Source description	Async DMA capable
31	Reserved	—	
32	TPM1	Channel 0	Yes
33	TPM1	Channel 1	Yes
34	TPM2	Channel 0	Yes
35	TPM2	Channel 1	Yes
36	Reserved	—	
37	Reserved	—	
38	Reserved	—	
39	Reserved	—	
40	ADC0	—	Yes
41	Reserved	—	
42	CMP0	—	Yes
43	Reserved	—	
44	Reserved	—	
45	DAC0	—	
46	Reserved	—	
47	Reserved	—	
48	Reserved	—	
49	Port control module	Port A	Yes
50	Reserved	—	
51	Port control module	Port C	Yes
52	Port control module	Port D	Yes
53	Reserved	—	
54	TPM0	Overflow	Yes
55	TPM1	Overflow	Yes
56	TPM2	Overflow	Yes
57	TSI	—	Yes
58	Reserved	—	
59	Reserved	—	
60	DMA MUX	Always enabled	
61	DMA MUX	Always enabled	
62	DMA MUX	Always enabled	
63	DMA MUX	Always enabled	

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

1.4.8.2 DMA transfers via PIT trigger

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

1.4.9 DMA Controller Configuration

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

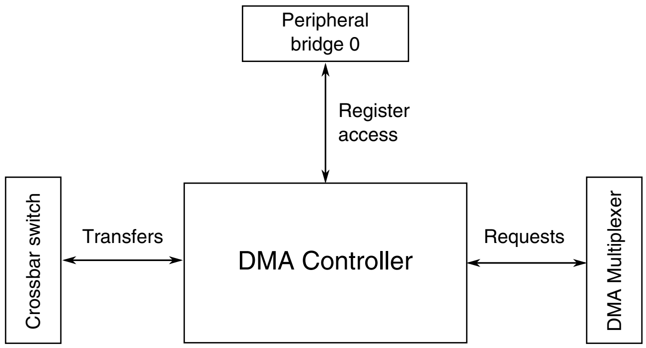


Figure 1-12. DMA Controller configuration

Table 1-21. Reference links to related information

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

1.4.10 Computer operating properly (COP) watchdog configuration

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

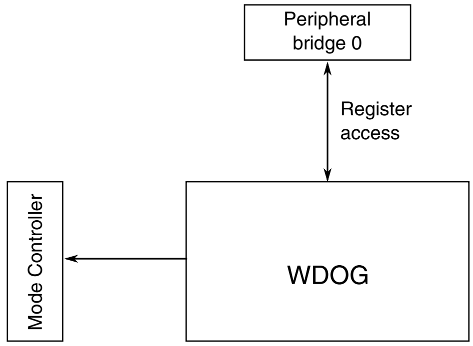


Figure 1-13. COP watchdog configuration

Table 1-22. Reference links to related information

Topic	Related module	Reference
Clocking	—	Clock distribution
Power management	—	Power management
Programming model	System integration module (SIM)	SIM

1.4.10.1 COP clocks

The multiple clock inputs for the COP are:

- 1 kHz clock
- bus clock

1.4.10.2 COP watchdog operation

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

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

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

SIM_COPC[COPCLKS] select the clock source used for the COP timer. The clock source options are either the bus clock or an internal 1 kHz clock source. With each clock source, the associated timeouts are controlled by SIM_COPC[COPT]. The following table summarizes the control functions of SIM_COPCTRL[COPCLKS] and SIM_COPC[COPT] fields. The COP watchdog defaults to operation from the 1 kHz clock source and the longest timeout is 2^{10} cycles.

Table 1-23. COP configuration options

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

After the bus clock source is selected, windowed COP operation is available by setting SIM_COPC[COPW]. In this mode, writes to SIM_SRVCOP to clear the COP timer must occur in the last 25% of the selected timeout period. A premature write immediately resets the chip. When the 1 kHz clock source is selected, windowed COP operation is not available.

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

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

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

If the 1 kHz clock source is selected, the COP counter is re-initialized to 0 upon entry to either Debug mode or Stop (including VLPS or LLS) mode. The counter begins from 0 upon exit from Debug mode or Stop mode.

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

1.4.10.3 Clock gating

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

1.5 Clock modules

1.5.1 MCG configuration

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

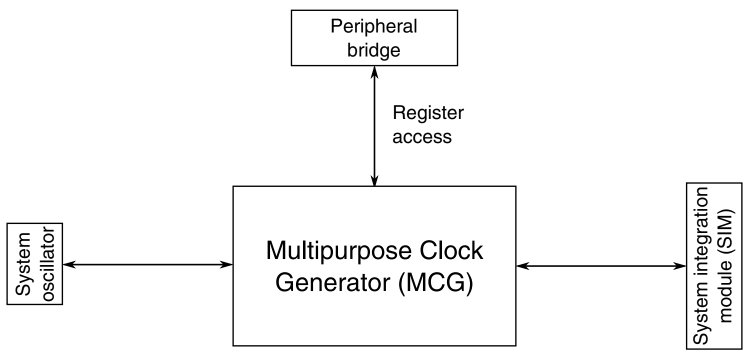


Figure 1-14. MCG configuration

Table 1-24. Reference links to related information

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

1.5.1.1 MCG FLL modes

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

1.5.2 OSC configuration

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

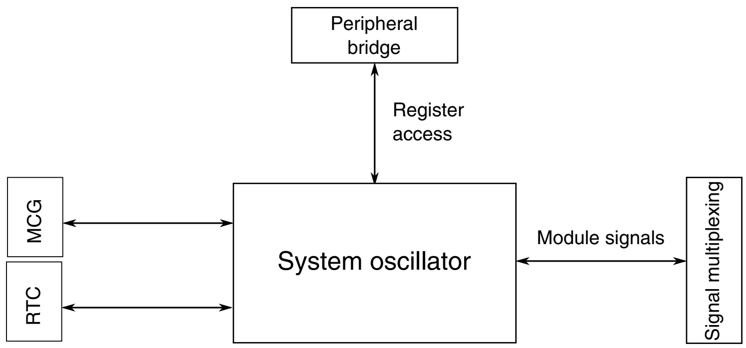


Figure 1-15. OSC configuration

Table 1-25. Reference links to related information

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

1.5.2.1 OSC modes of operation with MCG and RTC

The most common method of controlling the OSC block is through MCG_C1[CLKS] and the fields of MCG_C2 register to configure the oscillator frequency range, gain-mode, and for crystal or external clock operation. OSC_CR also provides control for enabling the OSC module and configuring internal load capacitors for the EXTAL and XTAL pins. See the OSC and MCG chapters for more details.

RTC_CR[OSCE] has overriding control over the MCG and OSC_CR enable functions. When RTC_CR[OSCE] is set, the OSC is configured for low frequency, low power and RTC_CR[SCxP] override OSC_CR[SCxP] to control the internal capacitance configuration. See the RTC chapter for more details.

1.6 Memories and memory interfaces

1.6.1 Flash memory configuration

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

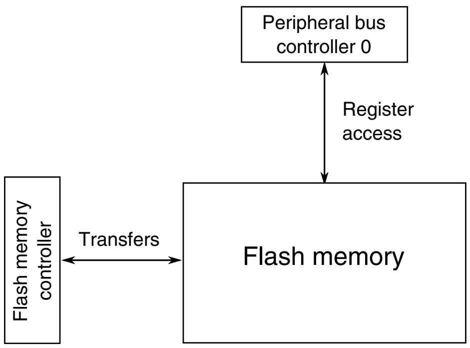


Figure 1-16. Flash memory configuration

Table 1-26. Reference links to related information

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

1.6.1.1 Flash memory sizes

The devices covered in this document contain 2 program flash blocks consisting of 1 KB sectors.

The amounts of flash memory for the devices covered in this document are:

Table 1-27. MKW01xxx flash memory size

Device	Program flash (KB)	Block 0 (P-Flash) address range	Block 1 (P-Flash) address range
MKW01xxx	128	0x0000_0000 – 0x0001_FFFF	0x0000_0000 – 0x0001_FFFF

1.6.1.2 Flash memory map

The flash memory and the flash registers are located at different base addresses as shown in the figure found here.

The base address for each is specified in System memory map.

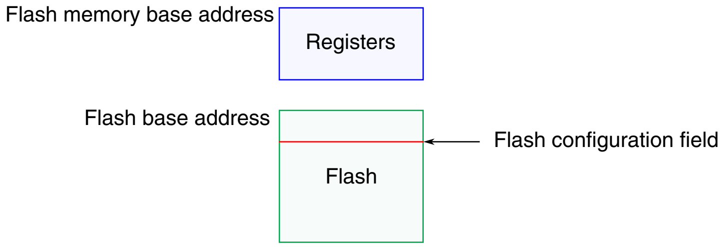


Figure 1-17. Flash memory map

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

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

1.6.1.3 Flash security

For information on how flash security is implemented on this device, see Chip Security.

1.6.1.4 Flash modes

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

1.6.1.5 Erase all flash contents

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

1.6.1.6 FTFA_FOPT register

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

1.6.2 Flash memory controller configuration

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

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

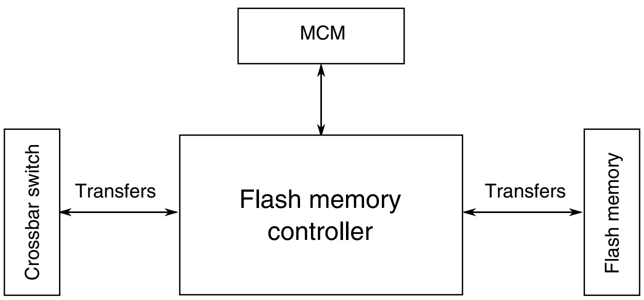


Figure 1-18. Flash memory controller configuration

Table 1-28. Reference links to related information

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

1.6.3 SRAM configuration

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

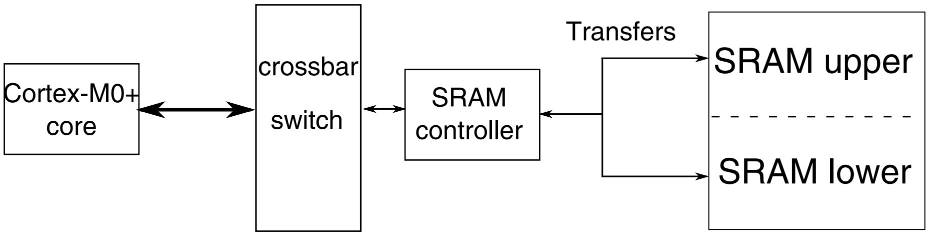


Figure 1-19. SRAM configuration

Table 1-29. Reference links to related information

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

1.6.3.1 SRAM sizes

This device contains SRAM which could be accessed by bus masters through the cross-bar switch. The amount of SRAM for the device covered in this document is shown in the following table.

Table 1-30. KW01xxx SRAM memory size

Device	SRAM (KB)
MKW01xxx	16

1.6.3.2 SRAM ranges

The on-chip SRAM is split into two ranges, 1/4 is allocated SRAM_L and 3/4 is allocated to SRAM_U.

The on-chip RAM is implemented such that the SRAM_L and SRAM_U ranges form a contiguous block in the memory map. 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.

Valid address ranges for SRAM_L and SRAM_U are then defined as:

- SRAM_L = [0x2000_0000–(SRAM_size/4)] to 0x1FFF_FFFF
- SRAM_U = 0x2000_0000 to [0x2000_0000+(SRAM_size*(3/4))-1]

This is illustrated in the following figure.

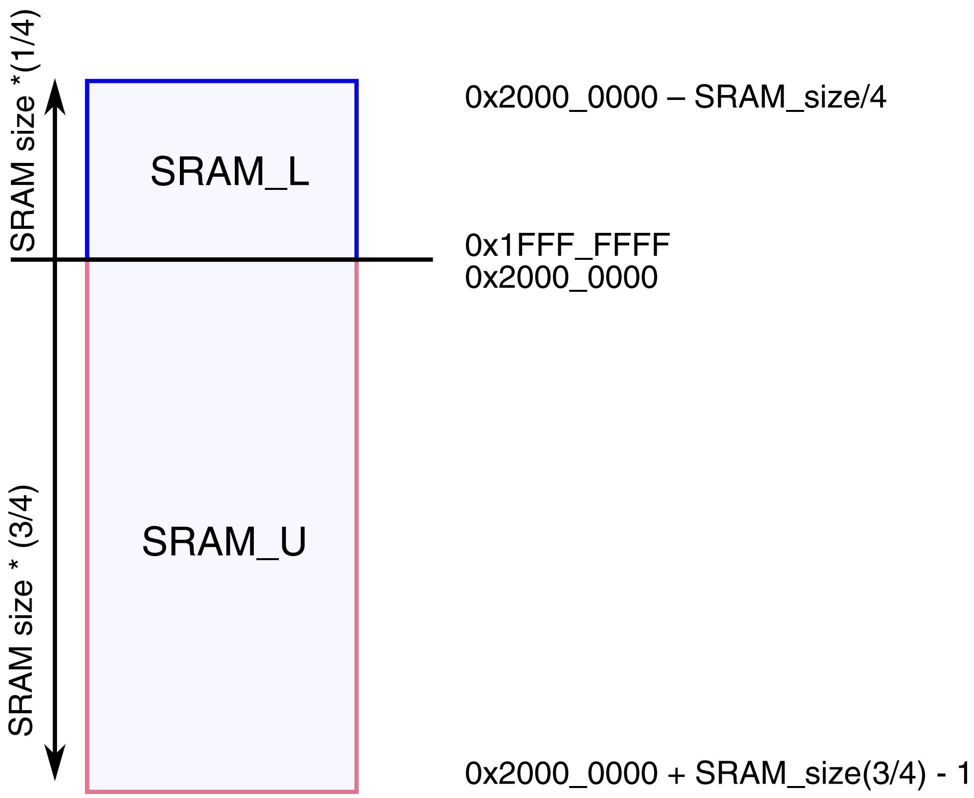


Figure 1-20. SRAM blocks memory map

For example, for a device containing 16 KB of SRAM, the ranges are:

- SRAM_L: 0x1FFF_F000 – 0x1FFF_FFFF
- SRAM_U: 0x2000_0000 – 0x2000_2FFF

1.6.3.3 SRAM retention in low power modes

The SRAM is retained down to VLLS3 mode. In VLLS1 and VLLS0, no SRAM is retained.

1.6.4 System Register File Configuration

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

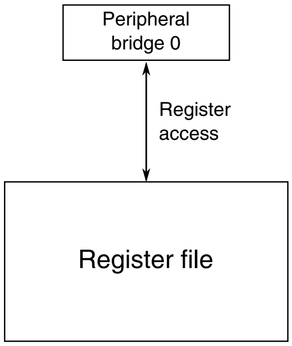


Figure 1-21. System Register file configuration

Table 1-31. Reference links to related information

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

1.6.4.1 System Register file

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

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

Base address of this register file is 0x4004_1000. (see Table "Peripheral bridge 0 slot assignment").

1.7 Analog

1.7.1 16-bit SAR ADC configuration

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

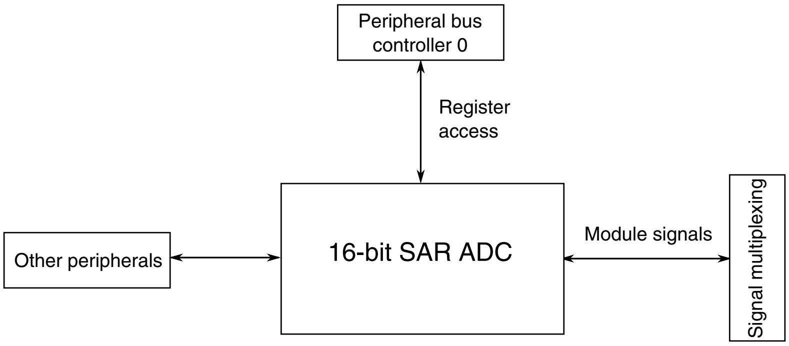


Figure 1-22. 16-bit SAR ADC configuration

Table 1-32. Reference links to related information

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

1.7.1.1 ADC instantiation information

This device contains one 16-bit successive approximation ADC.

The ADC supports both software and hardware triggers. The hardware trigger sources are listed in the Module-to-Module section.

The number of ADC channels present on the device is determined by the pinout of the specific device package and is shown in the following table.

Table 1-33. Number of MKW01xxx ADC channels

Device	Number of ADC channels
MKW01xxx	12

1.7.1.2 DMA Support on ADC

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

1.7.1.3 ADC0 connections/channel assignment

NOTE

As indicated by the following sections, each ADCx_DPx input and certain ADCx_DMx inputs may operate as single-ended ADC channels in single-ended mode.

1.7.1.3.1 ADC0 channel assignment

ADC channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	Reserved	Reserved
00001	DAD1	ADC0_DP1 and ADC0_DM1	ADC0_DP1/ADC0_SE1
00010	DAD2	ADC0_DP2 and ADC0_DM2	ADC0_DP2/ADC0_SE2
00011	DAD3	Reserved	Reserved
00100[1]	AD4a	Reserved	Reserved
00101	AD5a	Reserved	Reserved
00110	AD6a	Reserved	ADC0_DM2/ADC0_SE6a
00111	AD7a	Reserved	Reserved
00100	AD4b	Reserved	Reserved
00101	AD5b	Reserved	Reserved
00110	AD6b	Reserved	ADC0_SE6b
00111	AD7b	Reserved	ADC0_SE7b
01000	AD8	Reserved	ADC0_SE8
01001	AD9	Reserved	ADC0_SE9
01010	AD10	Reserved	Reserved
01011	AD11	Reserved	ADC0_SE11
01100	AD12	Reserved	ADC0_SE12
01101	AD13	Reserved	ADC0_SE13
01110	AD14	Reserved	ADC0_SE14
01111	AD15	Reserved	ADC0_SE15
10000	AD16	Reserved	Reserved
10001	AD17	Reserved	Reserved
10010	AD18	Reserved	Reserved
10011	AD19	Reserved	Reserved
10100	AD20	Reserved	Reserved
10101	AD21	Reserved	Reserved
10110	AD22	Reserved	Reserved
10111	AD23	Reserved	12-bit DAC0 Output/ ADC0_SE23

Table continues on the next page...

ADC channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
11000	AD24	Reserved	Reserved
11001	AD25	Reserved	Reserved
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff)[2]	Bandgap (S.E)
11100	AD28	Reserved	Reserved
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

1. ADCx_CFG2[MUXSEL] bit selects between ADCx_SEn channels a and b. Refer to MUXSEL description in ADC chapter for details.
2. This is the PMC bandgap 1V reference voltage. Prior to reading from this ADC channel, ensure that you enable the bandgap buffer by setting the PMC_REGSC[BGBE] bit. Refer to the device data sheet for the bandgap voltage (V_{BG}) specification.

1.7.1.4 ADC analog supply and reference connections

This device includes dedicated VDDA and VSSA pins.

This device contains separate VREFH and VREFL pins.

1.7.1.5 Alternate clock

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

NOTE

This clock option is only usable when OSCERCLK is in the MHz range. A system with OSCERCLK in the kHz range has the optional clock source below minimum ADC clock operating frequency.

1.7.2 CMP configuration

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

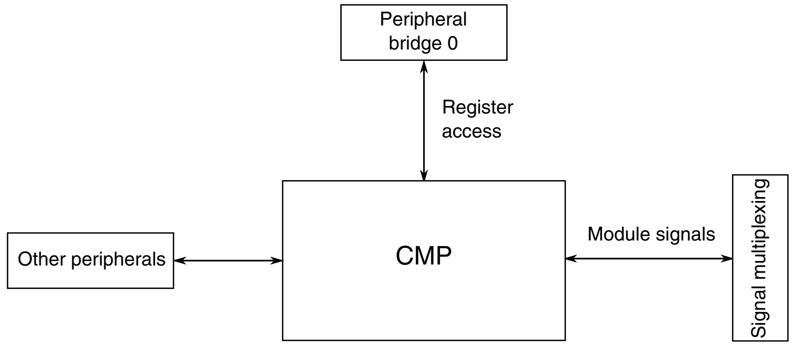


Figure 1-23. CMP configuration

Table 1-34. Reference links to related information

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

1.7.2.1 CMP instantiation information

The device includes one high-speed comparator and two 8-input multiplexers for both the inverting and non-inverting inputs of the comparator. Each CMP input channel connects to both muxes. Two of the channels are connected to internal sources, leaving resources to support up to 6 input pins. See the channel assignment table for a summary of CMP input connections for this device.

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

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

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

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

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

1.7.2.2 CMP input connections

The following table shows the fixed internal connections to the CMP.

Table 1-35. CMP input connections

CMP inputs	CMP0
IN0	CMP0_IN0
IN1	CMP0_IN1
IN2	CMP0_IN2
IN3	CMP0_IN3
IN4	12-bit DAC0 reference/ CMP0_IN4
IN5	CMP0_IN5
IN6	Bandgap[1]
IN7	6-bit DAC0 reference

1. This is the PMC bandgap 1V reference voltage. Prior to using as CMP input, ensure that you enable the bandgap buffer by setting PMC_REGSC[BGBE]. See the device data sheet for the bandgap voltage (V_{BG}) specification.

1.7.2.3 CMP external references

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

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

1.7.2.4 CMP trigger mode

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

- In Time Counter mode with prescaler enabled, the delay is 1/2 Prescaler output period.
- In Time Counter mode with prescaler bypassed, the delay is 1/2 Prescaler clock period.

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

1.7.3 12-bit DAC configuration

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

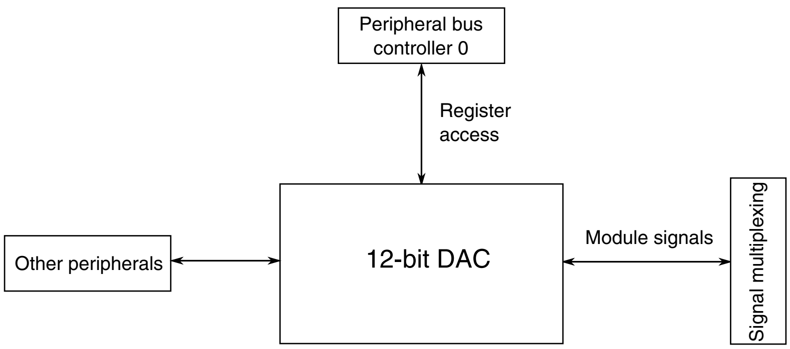


Figure 1-24. 12-bit DAC configuration

Table 1-36. Reference links to related information

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

1.7.3.1 12-bit DAC instantiation information

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

1.7.3.2 12-bit DAC output

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

1.7.3.3 12-bit DAC Analog Supply and Reference Connections

This device includes dedicated VDDA and VSSA pins.

This device contains separate VREFH and VREFL pins on 48-pin and higher devices. These pins are internally connected to VDDA and VSSA respectively, on 32-pin devices.

This device contains dedicated VREFH and VREFL pins in all packages.

1.7.3.4 12-bit DAC reference

VREFH is tied to VDDA internally.

Be aware that

NOTE

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

1.8 Timers

1.8.1 Timer/PWM module configuration

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

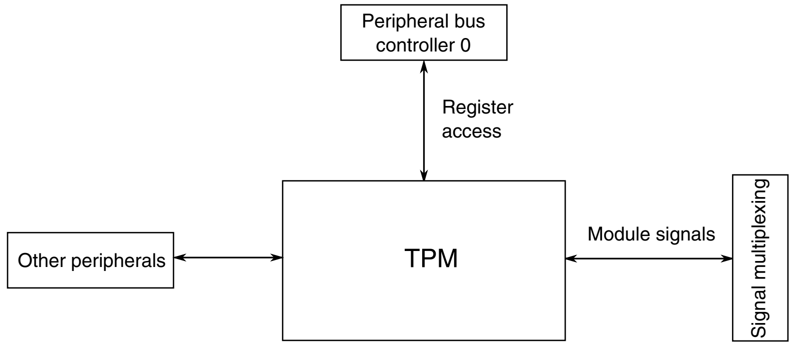


Figure 1-25. TPM configuration

Table 1-37. Reference links to related information

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

1.8.1.1 TPM instantiation information

This device contains three low power TPM modules (TPM). All TPM modules in the device are configured only as basic TPM function, do not support quadrature decoder function, and all can be functional in Stop/VLPS mode. The clock source is either external or internal in Stop/VLPS mode.

The following table shows how these modules are configured.

Table 1-38. TPM configuration

TPM instance	Number of channels	Features/usage
TPM0	6	Basic TPM,functional in Stop/VLPS mode
TPM1	2	Basic TPM,functional in Stop/VLPS mode
TPM2	2	Basic TPM,functional in Stop/VLPS mode

There are several connections to and from the TPMs in order to facilitate customer use cases. For complete details on the TPM module interconnects please refer to the Module-to-Module section.

1.8.1.2 Clock options

The TPM blocks are clocked from a single TPM clock that can be selected from OSCERCLK, MCGIRCLK, MCGPLLCLK/2, or MCGFLLCLK . The selected source is controlled by SIM_SOPT2[TPMSRC] and SIM_SOPT2[PLLFLLSEL].

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

1.8.1.3 Trigger options

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

Table 1-39. TPM trigger options

TPMx_CONF[TRGSEL]	Selected source
0000	External trigger pin input (EXTRG_IN)
0001	CMP0 output
0010	Reserved
0011	Reserved
0100	PIT trigger 0
0101	PIT trigger 1
0110	Reserved
0111	Reserved
1000	TPM0 overflow
1001	TPM1 overflow
1010	TPM2 overflow
1011	Reserved
1100	RTC alarm
1101	RTC seconds
1110	LPTMR trigger
1111	Reserved

1.8.1.4 Global timebase

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

1.8.1.5 TPM interrupts

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

1.8.2 PIT Configuration

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

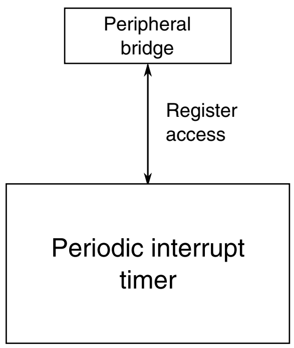


Figure 1-26. PIT configuration

Table 1-40. Reference links to related information

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

1.8.2.1 PIT/DMA periodic trigger assignments

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

Table 1-41. PIT channel assignments for periodic DMA triggering

PIT channel	DMA channel number
PIT Channel 0	DMA Channel 0
PIT Channel 1	DMA Channel 1

1.8.2.2 PIT/ADC triggers

PIT triggers are selected as ADCx trigger sources using the bits of SIM_SOPT7[ADCxTRGSEL]. For more details, see the SIM chapter.

1.8.2.3 PIT/TPM Triggers

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

1.8.2.4 PIT/DAC triggers

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

1.8.3 Low-power timer configuration

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

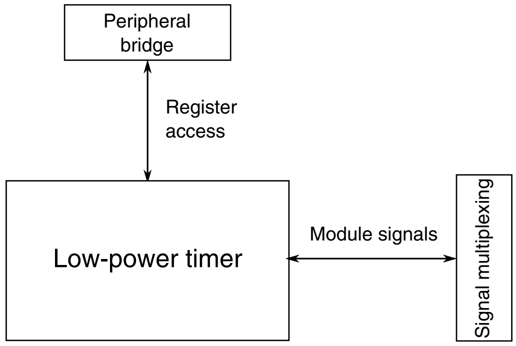


Figure 1-27. LPT configuration

Table 1-42. Reference links to related information

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

1.8.3.1 LPTMR instantiation information

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

The LPTMR can be clocked from the internal reference clock, the internal 1 kHz LPO, OSCERCLK, or an external 32.768 kHz crystal. In VLLS0 mode, the clocking option is limited to an external pin with the OSC configured for bypass (external clock) operation.

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

1.8.3.2 LPTMR pulse counter input options

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

LPTMR_CSR[TPS]	Pulse counter input number	Chip input
00	0	CMP0 output
01	1	LPTMR_ALT1 pin

Table continues on the next page...

LPTMR_CSR[TPS]	Pulse counter input number	Chip input
10	2	LPTMR_ALT2 pin
11	3	LPTMR_ALT3 pin

1.8.3.3 LPTMR prescaler/glitch filter clocking options

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

NOTE

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

LPTMR0_PSR[PCS]	Prescaler/glitch filter clock number	Chip clock
00	0	MCGIRCLK—internal reference clock (not available in LLS and VLLS modes)
01	1	LPO—1 kHz clock (not available in VLLS0 mode)
10	2	ERCLK32K (not available in VLLS0 mode when using 32 kHz oscillator)
11	3	OSCERCLK—external reference clock (not available in VLLS0 mode)

See Clock Distribution for more details on these clocks.

1.8.4 RTC configuration

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

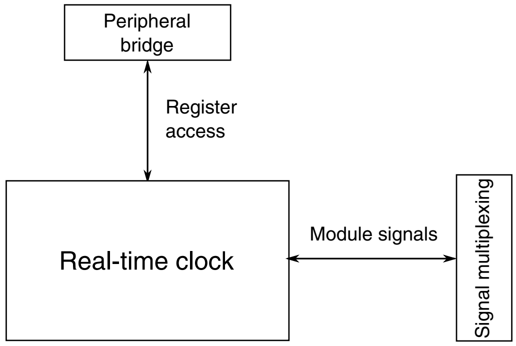


Figure 1-28. RTC configuration

Table 1-43. Reference links to related information

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

1.8.4.1 RTC Instantiation Information

RTC prescaler is clocked by ERCLK32K.

RTC is reset on POR Only.

RTC_CR[OSCE] can override the configuration of the System OSC, configuring the OSC for 32 kHz crystal operation in all power modes except VLLS0, and through any System Reset. When OSCE is enabled, the RTC also overrides the capacitor configurations.

1.8.4.2 RTC_CLKOUT options

RTC_CLKOUT pin can be driven either with the RTC 1 Hz output or with the OSCERCLK on-chip clock source. Control for this option is through SIM_SOPT2[RTCCLKOUTSEL].

When SIM_SOPT2[RTCCLKOUTSEL] = 0, the RTC 1 Hz clock is output is selected on the RTC_CLKOUT pin. When SIM_SOPT2[RTCCLKOUTSEL] = 1, OSCERCLK clock is output on the RTC_CLKOUT pin.

1.9 Communication interfaces

1.9.1 SPI configuration

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

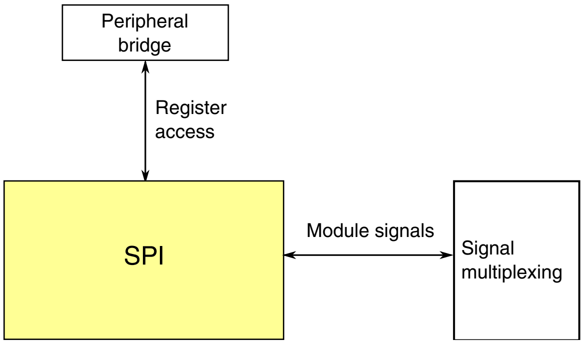


Figure 1-29. SPI configuration

Table 1-44. Reference links to related information

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

1.9.1.1 SPI instantiation information

This device contains two SPI module that supports 16-bit data length.

SPI1 includes a 4-deep FIFO.

SPI0 is clocked on the bus clock. SPI1 is clocked from the system clock. SPI1 is therefore disabled in "Partial Stop Mode".

The SPI supports DMA request and can operate in VLPS mode. When the SPI is operating in VLPS mode, it will operate as a slave.

SPI can wake the MCU from VLPS mode upon reception of SPI data in slave mode.

1.9.2 I2C configuration

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

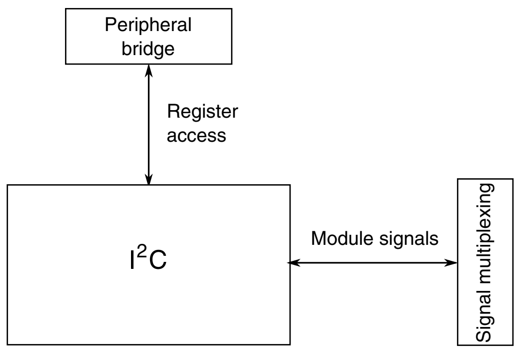


Figure 1-30. I2C configuration

Table 1-45. Reference links to related information

Topic	Related module	Reference
Full description	I^2C	I^2C
System memory map	—	System memory map
Clocking	—	Clock distribution
Power management	—	Power management
Signal multiplexing	Port control	Signal multiplexing

1.9.2.1 IIC instantiation information

This device has two IIC modules. IIC0 is clocked by the bus clock and IIC1 is clocked by the system clock. Clocking IIC1 at the faster system clock is needed to support standard IIC communication rates of 100 kbit/s in VLPR mode.

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

The digital glitch filter implemented in the IIC0 module, controlled by the I2C0_FLT[FLT] registers, is clocked from the bus clock and thus has filter granularity in bus clock cycle counts.

The digital glitch filter implemented in the IIC1 module, controlled by the I2C1_FLT[FLT] registers, is clocked from the system clock and thus has filter granularity in system clock cycle counts.

1.9.3 UART configuration

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

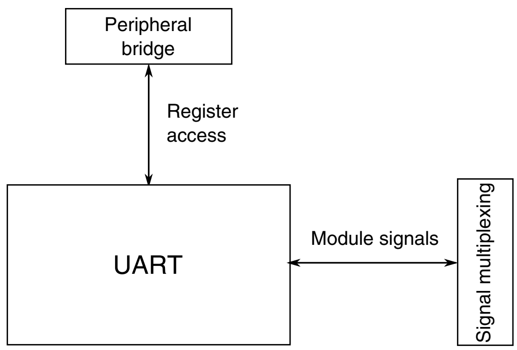


Figure 1-31. UART configuration

Table 1-46. Reference links to related information

Topic	Related module	Reference
Full description	UART1 and UART2	UART
Full description	UART0	UART
System memory map	—	System memory map
Clocking	—	Clock distribution
Power management	—	Power management
Signal multiplexing	Port control	Signal multiplexing

1.9.3.1 UART0 overview

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

This module supports LIN slave operation.

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

ISO7816 protocol is intended to be handled in software for this product. To support smart card reading, TxD pin can be configured as pseudo open drain for 1-wire half-duplex like ISO7816 communication via SIM_SOPT5[UART0ODE].

1.9.3.2 UART1 and UART2 Overview

This device contains two basic universal asynchronous receiver/transmitter (UART) modules with DMA function support. Generally, these modules are used in RS-232, RS-485, and other communications. This module supports LIN Slave operation.

1.10 Human-machine interfaces (HMI)

1.10.1 GPIO configuration

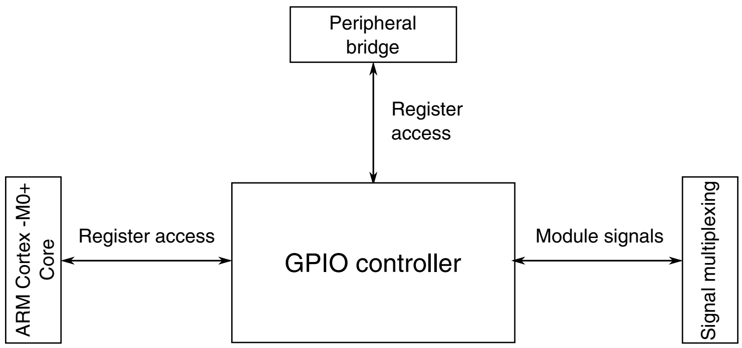


Figure 1-32. GPIO configuration

Table 1-47. Reference links to related information

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

1.10.1.1 GPIO instantiation information

The device includes a number of pins, PTB0, PTB1, and PTD6, PTD7 with high current drive capability. These pins can be used to drive LED or power MOSFET directly. The high drive capability applies to all functions which are multiplexed on these pins (UART, TPM, SPI, I2C, CLK_OUT...etc)

1.10.1.1.1 Pull devices and directions

The pull devices are enabled out of POR only on $\overline{\text{RESET}}$, $\overline{\text{NMI}}$ and respective SWD signals. Other pins can be enabled by writing to PORTx_PCRn[PE].

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

1.10.1.2 Port control and interrupt summary

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

Table 1-48. Ports summary

Feature	Port A	Port B	Port C	Port D	Port E
Pull select control	Yes	Yes	Yes	Yes	Yes
Pull select at reset	PTA0=Pull down, Others=Pull up	Pull up	Pull up	Pull up	Pull up
Pull enable control	Yes	Yes	Yes	Yes	Yes
Pull enable at reset	PTA0/PTA3/PTA4/ $\overline{\text{RESET\_b}}$=Enabled ; Others=Disabled	Disabled	Disabled	Disabled	Disabled
Slew rate enable control	Yes	Yes	Yes	Yes	Yes
Slew rate enable at reset	PTA3/PTA14/ PTA15/PTA16/ PTA17=Disabled; Others=Enabled	PTB10/PTB11/ PTB16/PTB17 = Disabled; Others=Enabled	PTC3/PTC4/PTC5/ PTC6/ PTC7=Disabled; Others=Enabled	PTD4/PTD5/PTD6/ PTD7=Disabled; Others=Enabled	PTE16/PTE17/ PTE18/ PTE19=Disabled; Others=Enabled
Passive filter enable control	PTA4 and $\overline{\text{RESET\_b}}$ only	No	No	No	No
Passive filter enable at reset	$\overline{\text{RESET\_b}}$=Enabled ; Others=Disabled	Disabled	Disabled	Disabled	Disabled
Open drain enable control[1]	No	No	No	No	No
Open drain enable at reset	Disabled	Disabled	Disabled	Disabled	Disabled
Drive strength enable control	No	PTB0/PTB1 only	No	PTD6/PTD7 only	No
Drive strength enable at reset	Disabled	Disabled	Disabled	Disabled	Disabled
Pin mux control	Yes	Yes	Yes	Yes	Yes
Pin mux at reset	PTA0/PTA3/ PTA4=ALT7; Others=ALT0	ALT0	ALT0	ALT0	ALT0
Lock bit	No	No	No	No	No

Table continues on the next page...

Table 1-48. Ports summary (continued)

Feature	Port A	Port B	Port C	Port D	Port E
Interrupt and DMA request	Yes	No	Yes	Yes	No
Digital glitch filter	No	No	No	No	No

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

1.10.1.3 GPIO accessibility in the memory map

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

1.10.2 TSI configuration

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

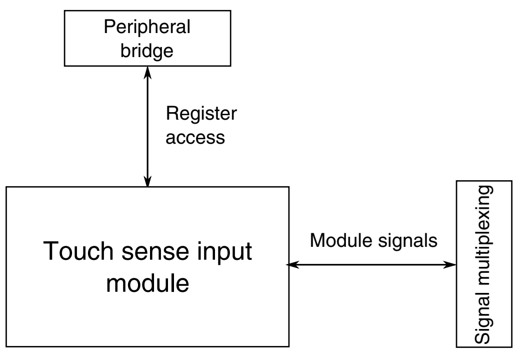


Figure 1-33. TSI configuration

Table 1-49. Reference links to related information

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

1.10.2.1 TSI instantiation information

This device includes one TSI module containing the channels as shown in the following table. In Stop, VLPS, LLS, and VLLSx modes any one channel can be enabled to be the wake-up source.

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

The number of TSI channels present on the device is determined by the pinout of the specific device package and is shown in the following table.

Table 1-50. Number of MKW01xxx TSI channels

Device	TSI channels
MKW01xxx	11

1.10.2.2 TSI Interrupts

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

Chapter 2
Memory Map

2.1 Introduction

This device contains various memories and memory-mapped peripherals which are located in a 4 GB memory space.

This chapter describes the memory and peripheral locations within that memory space.

2.2 System memory map

The table found here shows the high-level device memory map.

Table 2-1. System memory map

System 32-bit address range	Destination slave	Access
0x0000_0000–0x07FF_FFFF[1]	Program flash and read-only data (Includes exception vectors in first 196 bytes)	All masters
0x0800_0000–0x1FFF_DFFF	Reserved	—
0x1FFF_E000–0x1FFF_FFFF[2]	SRAM_L: Lower SRAM	All masters
0x2000_0000–0x2000_5FFF[2]	SRAM_U: Upper SRAM	All masters
0x2000_6000–0x3FFF_FFFF	Reserved	–
0x4000_0000–0x4007_FFFF	AIPS Peripherals	Cortex-M0+ core & DMA
0x4008_0000–0x400F_EFFF	Reserved	–
0x400F_F000–0x400F_FFFF	General-purpose input/output (GPIO)	Cortex-M0+ core & DMA
0x4010_0000–0x43FF_FFFF	Reserved	–
0x4400_0000–0x5FFF_FFFF	Bit Manipulation Engine (BME) access to AIPS Peripherals for slots 0-127[3]	Cortex-M0+ core
0x6000_0000–0xDFFF_FFFF	Reserved	–
0xE000_0000–0xE00F_FFFF	Private Peripherals	Cortex-M0+ core
0xE010_0000–0xEFFF_FFFF	Reserved	–

Table continues on the next page...

Table 2-1. System memory map (continued)

System 32-bit address range	Destination slave	Access
0xF000_0000–0xF000_0FFF	Micro Trace Buffer (MTB) registers	Cortex-M0+ core
0xF000_1000–0xF000_1FFF	MTB Data Watchpoint and Trace (MTBDWT) registers	Cortex-M0+ core
0xF000_2000–0xF000_2FFF	ROM table	Cortex-M0+ core
0xF000_3000–0xF000_3FFF	Miscellaneous Control Module (MCM)	Cortex-M0+ core
0xF000_4000–0xF7FF_FFFF	Reserved	–
0xF800_0000–0xFFFF_FFFF	IOPORT: GPIO (single cycle)	Cortex-M0+ core

1. The program flash always begins at 0x0000_0000 but the end of implemented flash varies depending on the amount of flash implemented for a particular device. See Flash memory sizes for details.
2. This range varies depending on SRAM sizes. See SRAM ranges for details.
3. Includes BME operations to GPIO at slot 15 (based at 0x4000_F000).

2.3 Flash memory map

The flash memory and the flash registers are located at different base addresses as shown in the figure found here.

The base address for each is specified in System memory map.

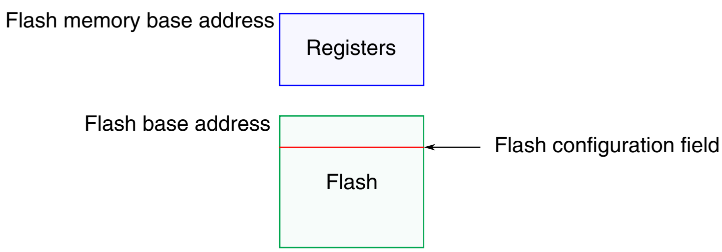


Figure 2-1. Flash memory map

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

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

2.3.1 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, the user software must load new values into the MCG trim registers.

Non-volatile byte address	Alternate IRC Trim Value
0x0000_03FC	Reserved
0x0000_03FD	Reserved
0x0000_03FE (bit 0)	SCFTRIM
0x0000_03FE (bit 4:1)	FCTRIM
0x0000_03FE (bit 6)	FCFTRIM
0x0000_03FF	SCTRIM

2.4 SRAM memory map

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

See SRAM ranges for details.

Access 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.

2.5 Bit Manipulation Engine

The Bit Manipulation Engine (BME) provides hardware support for atomic read-modify-write memory operations to the peripheral address space.

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

2.6 Peripheral bridge (AIPS-Lite) memory map

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

The three regions associated with this space are:

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

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

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

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

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

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

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

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

2.6.2 Peripheral bridge (AIPS-Lite) memory map

Table 2-2. Peripheral bridge 0 slot assignments

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

Table continues on the next page...

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

System 32-bit base address	Slot number	Module
0x4002_3000	35	—
0x4002_4000	36	—
0x4002_5000	37	—
0x4002_6000	38	—
0x4002_7000	39	—
0x4002_8000	40	—
0x4002_9000	41	—
0x4002_A000	42	—
0x4002_B000	43	—
0x4002_C000	44	—
0x4002_D000	45	—
0x4002_E000	46	—
0x4002_F000	47	I2S0
0x4003_0000	48	—
0x4003_1000	49	—
0x4003_2000	50	—
0x4003_3000	51	—
0x4003_4000	52	—
0x4003_5000	53	—
0x4003_6000	54	—
0x4003_7000	55	Periodic interrupt timers (PIT)
0x4003_8000	56	Timer'/PWM (TPM) 0
0x4003_9000	57	Timer'/PWM (TPM) 1
0x4003_A000	58	Timer'/PWM (TPM) 2
0x4003_B000	59	Analog-to-digital converter 0(ADC0)
0x4003_C000	60	—
0x4003_D000	61	Real-time clock (RTC)
0x4003_E000	62	—
0x4003_F000	63	DAC0
0x4004_0000	64	Low-power timer (LPTMR)
0x4004_1000	65	System register file
0x4004_2000	66	—
0x4004_3000	67	—
0x4004_4000	68	—
0x4004_5000	69	Touch sense interface (TSI)
0x4004_6000	70	—
0x4004_7000	71	SIM low-power logic
0x4004_8000	72	System integration module (SIM)
0x4004_9000	73	Port A multiplexing control

Table continues on the next page...

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

System 32-bit base address	Slot number	Module
0x4004_A000	74	Port B multiplexing control
0x4004_B000	75	Port C multiplexing control
0x4004_C000	76	Port D multiplexing control
0x4004_D000	77	Port E multiplexing control
0x4004_E000	78	—
0x4004_F000	79	—
0x4005_0000	80	—
0x4005_1000	81	—
0x4005_2000	82	—
0x4005_3000	83	—
0x4005_4000	84	—
0x4005_5000	85	—
0x4005_6000	86	—
0x4005_7000	87	—
0x4005_8000	88	—
0x4005_9000	89	—
0x4005_A000	90	—
0x4005_B000	91	—
0x4005_C000	92	—
0x4005_D000	93	—
0x4005_E000	94	—
0x4005_F000	95	—
0x4006_0000	96	—
0x4006_1000	97	—
0x4006_2000	98	—
0x4006_3000	99	—
0x4006_4000	100	Multi-purpose clock generator (MCG)
0x4006_5000	101	System oscillator (OSC)
0x4006_6000	102	I^2C0
0x4006_7000	103	I^2C1
0x4006_8000	104	—
0x4006_9000	105	—
0x4006_A000	106	UART0
0x4006_B000	107	UART1
0x4006_C000	108	UART2
0x4006_D000	109	—
0x4006_E000	110	—
0x4006_F000	111	—
0x4007_0000	112	—

Table continues on the next page...

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

System 32-bit base address	Slot number	Module
0x4007_1000	113	—
0x4007_2000	114	—
0x4007_3000	115	Analog comparator (CMP) / 6-bit digital-to-analog converter (DAC)
0x4007_4000	116	—
0x4007_5000	117	—
0x4007_6000	118	SPI0
0x4007_7000	119	SPI1
0x4007_8000	120	—
0x4007_9000	121	—
0x4007_A000	122	—
0x4007_B000	123	—
0x4007_C000	124	Low-leakage wakeup unit (LLWU)
0x4007_D000	125	Power management controller (PMC)
0x4007_E000	126	System mode controller (SMC)
0x4007_F000	127	Reset control module (RCM)
0x400F_F000	128	GPIO controller

2.6.3 Modules restricted access in user mode

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

2.7 Private Peripheral Bus (PPB) memory map

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

Table 2-3. PPB memory map

System 32-bit Address Range	Resource	Additional Range Detail	Resource
0xE000_0000–0xE000_DFFF	Reserved		

Table continues on the next page...

Table 2-3. PPB memory map (continued)

System 32-bit Address Range	Resource	Additional Range Detail	Resource
0xE000_E000–0xE000_EFFF	System Control Space (SCS)	0xE000_E000–0xE000_E00F	Reserved
		0xE000_E010–0xE000_E0FF	SysTick
		0xE000_E100–0xE000_ECFF	NVIC
		0xE000_ED00–0xE000_ED8F	System Control Block
		0xE000_ED90–0xE000_EDEF	Reserved
		0xE000_EDF0–0xE000_EEFF	Debug
		0xE000_EF00–0xE000_EFFF	Reserved
0xE000_F000–0xE00F_EFFF	Reserved		
0xE00F_F000–0xE00F_FFFF	Core ROM Space (CRS)		

Chapter 3
Clock Distribution

3.1 Introduction

The information found here presents the clock architecture for the device, the overview of the clocks, and includes a terminology section.

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

See AN4503: Power Management for Kinetis and ColdFire+ MCUs for further details on the system clocks and their use.

3.2 Programming model

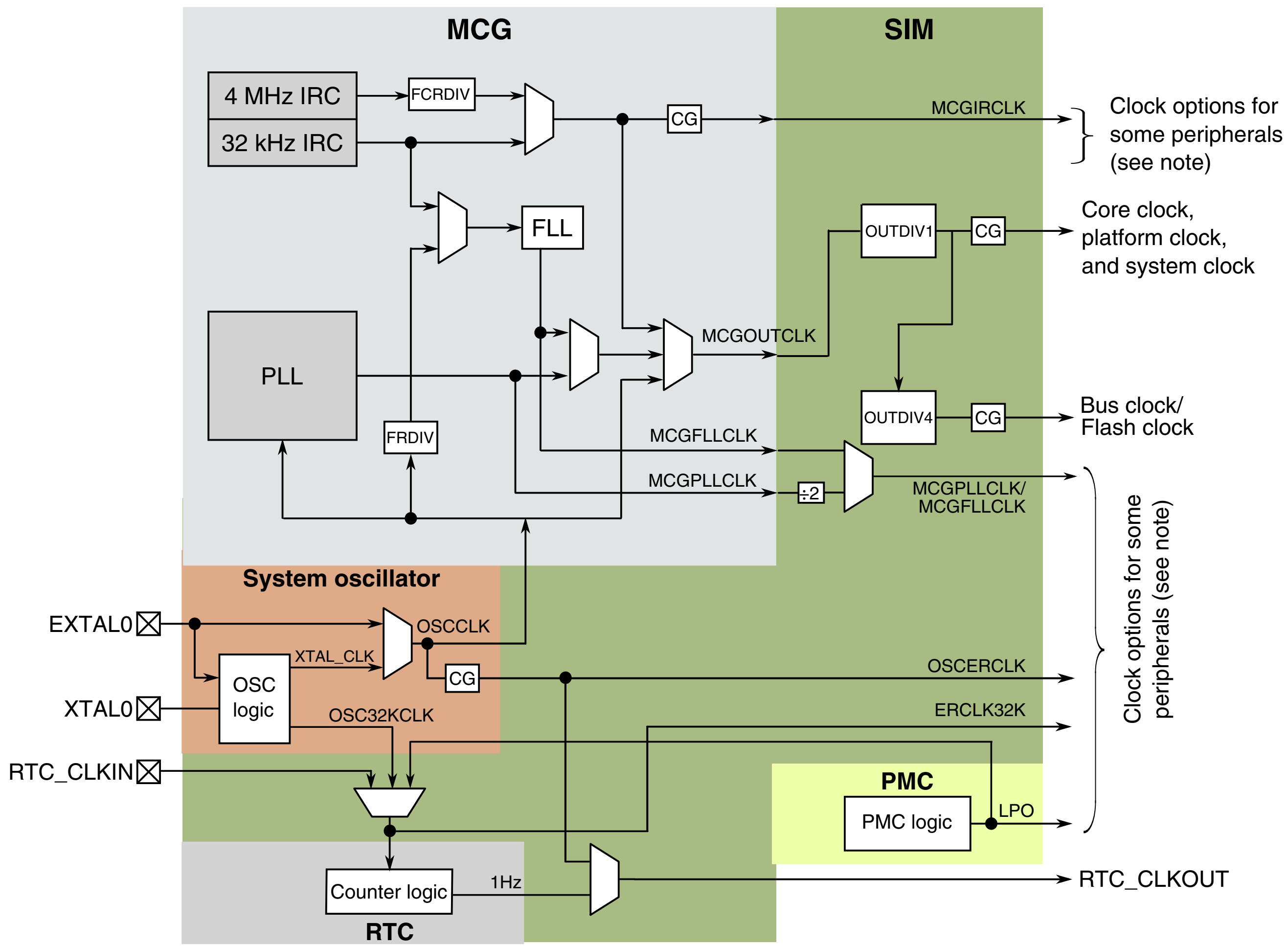
The selection and multiplexing of system clock sources is controlled and programmed via the MCG module.

The setting of clock dividers and module clock gating for the system are programmed via the SIM module. Refer to the MCG and SIM sections for detailed register and bit descriptions.

3.3 High-level device clocking diagram

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

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



CG — Clock gate
Note: See subsequent sections for details on where these clocks are used.

Figure 3-1. Clocking diagram

3.4 Clock definitions

This table describes the clocks in the block diagram.

Clock name	Description
Core clock	MCGOUTCLK divided by OUTDIV1 Clocks the ARM Cortex-M0+ core.
Platform clock	MCGOUTCLK divided by OUTDIV1 Clocks the crossbar switch and NVIC.
System clock	MCGOUTCLK divided by OUTDIV1 Clocks the bus masters directly.
Bus clock	System clock divided by OUTDIV4. Clocks the bus slaves and peripherals.
Flash clock	Flash memory clock On this device, it is the same as Bus clock.
MCGIRCLK	MCG output of the slow or fast internal reference clock
MCGOUTCLK	MCG output of either IRC, MCGFLLCLK, MCGPLLCLK, or MCG's external reference clock that sources the core, system, bus, and flash clock.
MCGFLLCLK	MCG output of the FLL MCGFLLCLK or MCGPLLCLK may clock some modules. In addition, this clock is used for UART0 and TPM modules.
MCGPLLCLK	MCG output of the PLL. MCGFLLCLK or MCGPLLCLK may clock some modules. In addition, this clock is used for UART0 and TPM modules.
OSCCLK	System oscillator output of the internal oscillator or sourced directly from EXTAL. Used as MCG external reference clock.
OSCERCLK	System oscillator output sourced from OSCCLK that may clock some on-chip modules
OSC32KCLK	System oscillator 32 kHz output
ERCLK32K	Clock source for some modules that is chosen as OSC32KCLK or RTC_CLKIN
LPO	PMC 1 kHz output

3.4.1 Device clock summary

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

Table 3-1. Clock summary

Clock name	Run mode clock frequency	VLPR mode clock frequency	Clock source	Clock is disabled when…
MCGOUTCLK	Up to 100 MHz	Up to 4 MHz	MCG	In all stop modes except for partial stop modes and during PLL locking when MCGOUTCLK derived from PLL.
MCGFLLCLK	Up to 48 MHz	N/A	MCG	MCG clock controls are not enabled and in all stop modes
MCGPLLCLK	Up to 100 MHz	N/A	MCG	MCG clock controls do not enable, in Stop mode but PLLSTEN=0, or in VLPS, LLS and VLLSx modes
Core clock	Up to 48 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all wait and stop modes
Platform clock	Up to 48 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all stop modes
System clock	Up to 48 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all stop modes and Compute Operation
Bus clock	Up to 24 MHz	Up to 1 MHz [1]	MCGOUTCLK clock divider	In all stop modes except for partial STOP2 mode, and Compute Operation
SWD Clock	Up to 24 MHz	Up to 1 MHz	SWD_CLK pin	In all stop modes
Flash clock	Up to 24 MHz	Up to 1 MHz in BLPE Up to 800 kHz in BLPI	MCGOUTCLK clock divider	In all stop modes except for partial STOP2 mode
Internal reference (MCGIRCLK)	30–40 kHz Slow IRC or 4 MHz Fast IRC	4 MHz Fast IRC only	MCG	MCG_C1[IRCLKEN] cleared, Stop/VLPS mode and MCG_C1[IREFSTEN] cleared, or LLS/VLLS mode
External reference (OSCERCLK)	Up to 48 MHz (bypass), 30–40 kHz or 3-32 MHz (crystal)	Up to 16 MHz (bypass), 30–40 kHz (low-range crystal) or 3-16 MHz (high-range crystal)	System OSC	OSC_CR[ERCLKEN] cleared, or Stop mode and OSC_CR[EREFSTEN] cleared or VLLS0 and oscillator not in external clock mode.

Table continues on the next page...

Table 3-1. Clock summary (continued)

Clock name	Run mode clock frequency	VLPR mode clock frequency	Clock source	Clock is disabled when…
External reference 32kHz (ERCLK32K)	30–40 kHz	30–40 kHz	System OSC , or RTC_CLKIN	OSC_CR[ERCLKEN] cleared and RTC_CR[OSCE] cleared or VLLS0 and oscillator not in external clock mode.
RTC_CLKOUT	RTC 1Hz, OSCERCLK	RTC 1Hz, OSCERCLK	RTC 1Hz, OSCERCLK	Clock is disabled in LLS and VLLSx modes
LPO	1 kHz	1 kHz	PMC	in VLLS0
TPM clock	Up to 48 MHz	Up to 8 MHz	MCGIRCLK, MCGPLLCLK/2, MCGFLLCLK, or OSCERCLK	SIM_SOPT2[TPMSRC]=00 or selected clock source disabled.
UART0 clock	Up to 48 MHz	Up to 8 MHz	MCGIRCLK, MCGPLLCLK/2, MCGFLLCLK, or OSCERCLK	SIM_SOPT2[UART0SRC]=00 or selected clock source disabled.
I2S master clock	Up to 25 MHz	Up to 16 MHz	System clock, MCGPLLCLK, OSCERCLK or I2S_CLKIN	I^2S is disabled

1. If in BLPI mode, where clocking is derived from the fast internal reference clock, the Bus clock and flash clock frequency needs to be limited to 800 kHz if executing from flash.

3.5 Internal clocking requirements

The clock dividers are programmed via the CLKDIV registers of the SIM module.

The following requirements must be met when configuring the clocks for this device:

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

The following is a common clock configuration for this device:

Clock	Frequency
Core clock	48 MHz
Platform clock	48 MHz
System clock	48 MHz
Bus clock	24 MHz
Flash clock	24 MHz

3.5.1 Clock divider values after reset

Each clock divider is programmed via the CLKDIV1 registers of the SIM module. Two bits in the flash memory's FTFA_FOPT register control the reset value of the core clock, system clock, bus clock, and flash clock dividers as shown in this table.

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

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

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

3.5.2 VLPR mode clocking

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

- the core/system clocks are less than or equal to 4 MHz, and
- the bus and flash clocks are less than or equal to 1 MHz

NOTE

When the MCG is in BLPI and clocking is derived from the Fast IRC, the clock divider controls (MCG_SC[FCRDIV], SIM_CLKDIV1[OUTDIV1], and SIM_CLKDIV1[OUTDIV4]) must be programmed such that the resulting flash clock nominal frequency is 800 kHz or less. In this case, one example of correct configuration is MCG_SC[FCRDIV] = 000b, SIM_CLKDIV1[OUTDIV1] = 0000b, and SIM_CLKDIV1[OUTDIV4] = 100b, resulting in a divide-by-5 setting.

3.6 Clock gating

The clock to each module can be individually gated on and off using bits of the SCGC*x* registers of the SIM module.

These bits are cleared after any reset, which disables the clock to the corresponding module to conserve power. Prior to initializing a module, set the corresponding bit in the SCGC*x* register to enable the clock. Before turning off the clock, make sure to disable the module.

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

3.7 Module clocks

The table found here summarizes the clocks associated with each module.

Table 3-2. Module clocks

Module	Bus interface clock	Internal clocks	I/O interface clocks
Core modules			
ARM Cortex-M0+ core	Platform clock	Core clock	—
NVIC	Platform clock	—	—
DAP	Platform clock	—	SWD_CLK
System modules			
DMA	System clock	—	—
DMA Mux	Bus clock	—	—
Port control	Bus clock	—	—
Crossbar Switch	Platform clock	—	—
Peripheral bridges	System clock	Bus clock	—
LLWU, PMC, SIM, RCM	Bus clock	LPO	—
Mode controller	Bus clock	—	—
MCM	Platform clock	—	—
Watchdog timer	Bus clock	LPO	—

Table continues on the next page...

Table 3-2. Module clocks (continued)

Module	Bus interface clock	Internal clocks	I/O interface clocks
Clocks			
MCG	Bus clock	MCGOUTCLK, MCGPLLCLK, MCGFLLCLK, MCGIRCLK, OSCERCLK	—
OSC	Bus clock	OSCERCLK	—
Memory and memory interfaces			
Flash Controller	Platform clock	Flash clock	—
Flash memory	Flash clock	—	—
Analog			
ADC	Bus clock	OSCERCLK	—
CMP	Bus clock	—	—
DAC	Bus clock	—	—
Timers			
TPM	Bus clock	TPM clock	TPM_CLKIN0, TPM_CLKIN1
PIT	Bus clock	—	—
LPTMR	Bus clock	LPO, OSCERCLK, MCGIRCLK, ERCLK32K	—
RTC	Bus clock	ERCLK32K	RTC_CLKOUT
Communication interfaces			
SPI0	Bus clock	—	SPI0_SCK
SPI1	System clock	—	SPI1_SCK
I^2C0	Bus clock	—	I2C0_SCL
I^2C1	System clock	—	I2C1_SCL
UART0	Bus clock	UART0 clock	—
UART1 , UART2	Bus clock	—	—
I^2S	Bus clock	I^2S master clock	I2S_TX_BCLK, I2S_RX_BCLK
Human-machine interfaces			
GPIO	Platform clock	—	—
TSI	Bus clock	—	—

3.7.1 PMC 1-kHz LPO clock

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

3.7.2 COP clocking

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

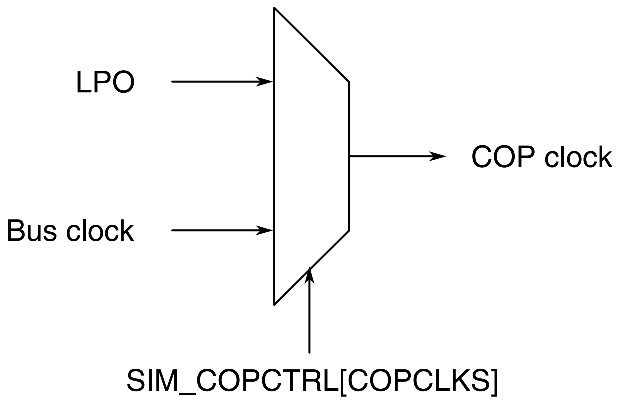


Figure 3-2. COP clock generation

3.7.3 RTC clocking

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

NOTE

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

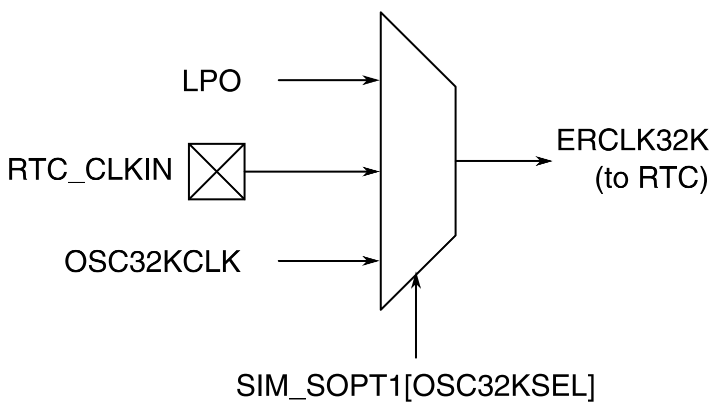


Figure 3-3. RTC clock generation

3.7.4 LPTMR clocking

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

NOTE

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

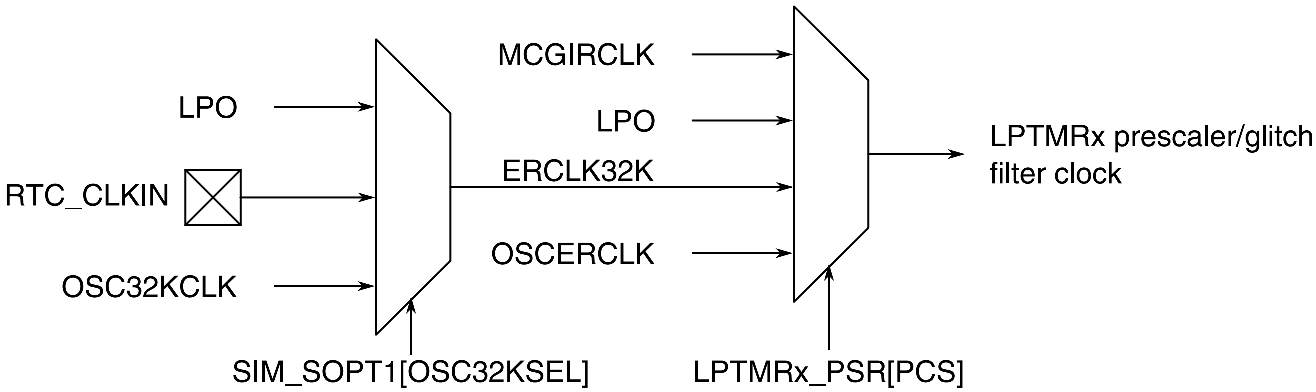


Figure 3-4. LPTMRx prescaler/glitch filter clock generation

3.7.5 TPM clocking

The counter for the TPM modules has a selectable clock as shown in the following figure.

NOTE

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

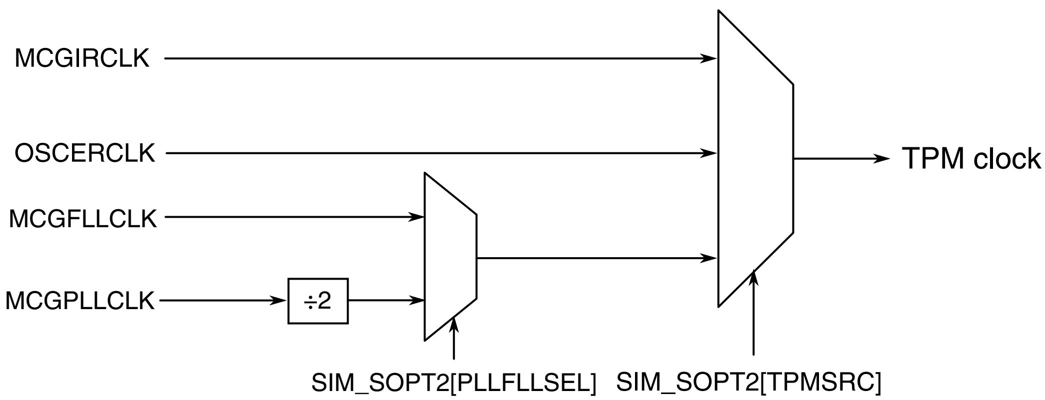


Figure 3-5. TPM clock generation

3.7.6 SPI clocking

SPI0 is clocked from the bus clock. That is, the SPI0 module clock is connected to the chip-level bus clock.

SPI1 is clocked from the system clock. That is, the SPI1 module clock is connected to the chip-level system clock. SPI1 is therefore disabled in "Partial Stop Mode".

3.7.7 I2C clocking

I2C0 is clocked from the bus clock.

I2C1 is clocked from the system clock. Clocking I2C1 at the faster system clock is needed to support standard I2C communication rates of 100 kbit/s in VLPR mode.

3.7.8 UART clocking

The UART0 module has a selectable clock as shown in the following figure. UART1 and UART2 modules operate from the bus clock.

NOTE

The chosen clock must remain enabled if the UART0 is to continue operating in all required low-power modes.

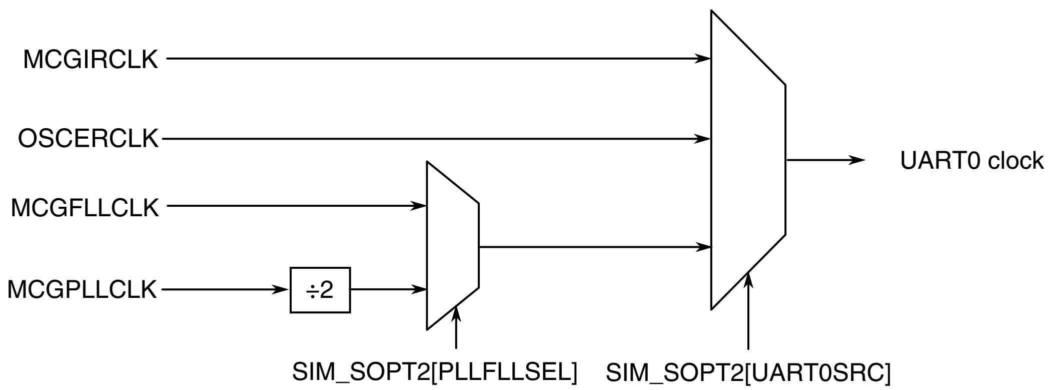


Figure 3-6. UART0 clock generation

3.7.9 I²S/SAI clocking

The audio master clock (MCLK) is used to generate the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The audio master clock can also be output to or input from a pin. The transmitter and receiver have the same audio master clock inputs.

Each SAI peripheral can control the input clock selection, pin direction and divide ratio of one audio master clock.

The I²S/SAI transmitter and receiver support asynchronous bit clocks (BCLKs) that can be generated internally from the audio master clock or supplied externally. The module also supports the option for synchronous operation between the receiver and transmitter.

The transmitter and receiver can independently select between the bus clock and the audio master clock to generate the bit clock.

The MCLK and BCLK source options appear in the following figure.

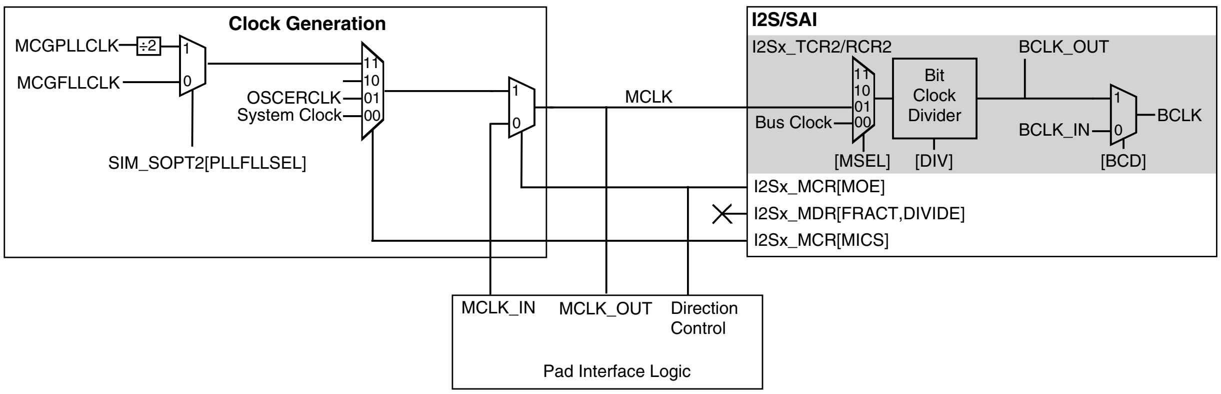


Figure 3-7. I²S/SAI clock generation

Chapter 4
Reset and Boot

4.1 Introduction

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

Table 4-1. Reset sources

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

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

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

4.2 Reset

The information found here discusses basic reset mechanisms and sources.

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

4.2.1 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_{LVDL}). The POR and LVD fields in the Reset Status Register are set following a POR.

4.2.2 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 program counter (PC) from vector-table offset 4
- Link register (LR) is set to 0xFFFF_FFFF.

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

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

- SWD_CLK in pulldown (PD)
- SWD_DIO in pullup (PU)

4.2.2.1 External pin reset ($\overline{\text{RESET}}$)

This pin is open drain and has an internal pullup device. Asserting $\overline{\text{RESET}}$ wakes the device from any mode.

The $\overline{\text{RESET}}$ pin can be disabled by programming RESET_PIN_CFG option bit to 0. When this option is selected, there could be a short period of contention during a POR ramp where the device drives the pin-out low prior to establishing the setting of this option and releasing the reset function on the pin.

4.2.2.1.1 $\overline{\text{RESET}}$ pin filter

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

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

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

4.2.2.2 Low-voltage detect (LVD)

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

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

4.2.2.3 Computer operating properly (COP) watchdog timer

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

4.2.2.4 Low leakage wakeup (LLWU)

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

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

NOTE

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

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

The MCG module supports an external reference clock.

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

NOTE

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

4.2.2.6 MCG loss-of-lock (LOL) reset

The MCG includes a PLL loss-of-lock detector. The detector is enabled when configured for PEE and lock has been achieved. If MCG_C8[LOLRE] is set and the PLL Lock Status field (MCG_S[LOLS0]) becomes set, the MCU resets. RCM_SRS0[LOL] is set to indicate this reset source.

NOTE

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

4.2.2.7 Stop mode acknowledge error (SACKERR)

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

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

4.2.2.8 Software reset (SW)

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

4.2.2.9 Lockup reset (LOCKUP)

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

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

4.2.2.10 MDM-AP system reset request

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

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

4.2.3 MCU resets

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

4.2.3.1 POR Only

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

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

4.2.3.2 Chip POR not VLLS

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

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

4.2.3.3 Chip POR

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

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

4.2.3.4 Chip Reset not VLLS

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

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

4.2.3.5 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).

4.2.3.6 Chip Reset

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

4.2.4 $\overline{\text{RESET}}$ pin

For all reset sources except a VLLS Wakeup that does not occur via the $\overline{\text{RESET}}$ pin, the $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ pin is released, and the internal Chip Reset negates after the $\overline{\text{RESET}}$ pin is pulled high. Keeping the $\overline{\text{RESET}}$ pin asserted externally delays the negation of the internal Chip Reset.

The $\overline{\text{RESET}}$ pin can be disabled by programming FTFA_FOPT[RESET_PIN_CFG] option bit to 0 (See Table 4-2). When this option is selected, there could be a short period of contention during a POR ramp where the device drives the pinout low prior to establishing the setting of this option and releasing the reset function on the pin.

4.2.5 Debug resets

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

4.2.5.1 Resetting the Debug subsystem

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

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

- SW-DP
- AHB-AP
- MDM-AP (MDM control and status registers)

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

- CM0+ core (core debug registers: DHCSR, DCRSR, DCRDR, DEMCR)
- BPU

- DWT
- NVIC
- Crossbar bus switch[1]
- AHB-AP[1]
- Private peripheral bus[1]

4.3 Boot

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

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

4.3.1 Boot sources

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

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

4.3.2 FOPT boot options

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

1. CDBGRSTREQ does not affect AHB resources so that debug resources on the private peripheral bus are available during System Reset.

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

Table 4-2. Flash Option Register (FTFA_FOPT) bit definitions

Bit Num	Field	Value	Definition
7-6	Reserved	Reserved for future expansion.	
5	FAST_INIT	Selects initialization speed on POR, VLLSx, and any system reset .	
		0	Slower initialization: The flash initialization will be slower with the benefit of reduced average current during this time. The duration of the recovery will be controlled by the clock divider selection determined by the LPBOOT setting.
		1	Fast Initialization: The flash has faster recoveries at the expense of higher current during these times.
3	RESET_PIN_CFG	Enables/disables control for the RESET pin.	
		0	$\overline{\mathrm{RESET\_b}}$ 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 pinout low prior to establishing the setting of this option and releasing the reset function on the pin. This bit is preserved through system resets and low-power modes. When $\overline{\mathrm{RESET\_b}}$ 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	$\overline{\mathrm{RESET\_b}}$ pin is dedicated. The port is configured with pullup enabled, open drain, passive filter enabled.
2	NMI_DIS	Enables/disables control for the NMI function.	
		0	NMI interrupts are always blocked. The associated pin continues to default to $\overline{\mathrm{NMI\_b}}$ pin controls with internal pullup enabled. When $\overline{\mathrm{NMI\_b}}$ pin function is disabled, it cannot be used as a source for low-power mode wake-up.
		1	$\overline{\mathrm{NMI\_b}}$ pin/interrupts reset default to enabled.
1	Reserved	Reserved for future expansion.	
4,0	LPBOOT	Controls the reset value of OUTDIV1 value in SIM_CLKDIV1 register. Larger divide value selections produce lower average power consumption during POR, VLLSx recoveries and reset sequencing and after reset exit. The recovery times are also extended if the FAST_INIT option is not selected.	
		00	Core and system clock divider (OUTDIV1) is 0x7 (divide by 8).
		01	Core and system clock divider (OUTDIV1) is 0x3 (divide by 4).
		10	Core and system clock divider (OUTDIV1) is 0x1 (divide by 2).
		11	Core and system clock divider (OUTDIV1) is 0x0 (divide by 1).

4.3.3 Boot sequence

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

1. A system reset is held on internal logic, the $\overline{\text{RESET}}$ pin is driven out low, and the MCGis enabled in its default clocking mode.
2. Required clocks are enabled (system clock, flash clock, and any bus clocks that do not have clock gate control reset to disabled).
3. The system reset on internal logic continues to be held, but the Flash Controller is released from reset and begins initialization operation while the Reset Control logic continues to drive the $\overline{\text{RESET}}$ pin out low.
4. Early in reset sequencing, the NVM option byte is read and stored to the FOPT register of the Flash Memory module (FTFA_FOPT). If the bits associated with FTFA_FOPT[LPBOOT] are programmed for an alternate clock divider reset value, the system/core clock is switched to a slower clock speed. If FTFA_FOPT[FAST_INIT] is programmed clear, the flash initialization switches to slower clock resulting longer recovery times.
5. When flash Initialization completes, the $\overline{\text{RESET}}$ pin is released. If $\overline{\text{RESET}}$ continues to be asserted (an indication of a slow rise time on the $\overline{\text{RESET}}$ pin or external drive in low), the system continues to be held in reset. Once the $\overline{\text{RESET}}$ pin is detected high, the core clock is enabled and the system is released from reset.
6. When the system exits reset, the processor sets up the stack, program counter (PC), and link register (LR). The processor reads the start SP (SP_main) from vector-table offset 0. The core reads the start PC from vector-table offset 4. LR is set to 0xFFFF_FFFF. The next sequence of events depends on the $\overline{\text{NMI}}$ input and FTFA_FOPT[NMI_DIS] (See Table 4-2) :
   - If the $\overline{\text{NMI}}$input is high or the NMI function is disabled in FTFA_FOPT, the CPU begins execution at the PC location.
   - If the $\overline{\text{NMI}}$ input is low and the NMI function is enabled in FTFA_FOPT, this results in an NMI interrupt. The processor executes an Exception Entry and reads the NMI interrupt handler address from vector-table offset 8. The CPU begins execution at the NMI interrupt handler.

   /

Subsequent system resets follow this same reset flow.

Chapter 5
Power Management

5.1 Introduction

Information about the various chip power modes and functionality of the individual modules in these modes can be found here.

See AN4503: Power Management for Kinetis and ColdFire+ MCUs for further details on power management techniques.

5.2 Clocking modes

Information found here describes the various clocking modes supported on this device.

5.2.1 Partial Stop

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

When configured for PSTOP2, only the core and system clocks are gated and the bus clock remains active. The bus masters and bus slaves clocked by the system clock enter Stop mode, but the bus slaves clocked by bus clock remain in Run (or VLP Run) mode. The clock generators in the MCG and the on-chip regulator in the PMC also remain in Run (or VLP Run) mode. Exit from PSTOP2 can be initiated by a reset, an asynchronous interrupt from a bus master or bus slave clocked by the system clock, or a synchronous interrupt from a bus slave clocked by the bus clock. If configured, a DMA request (using the asynchronous DMA wakeup) can also be used to exit Partial Stop for the duration of a DMA transfer before the device is transitioned back into PSTOP2.

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

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

5.2.2 DMA Wakeup

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 MCG, enabling the system and bus clocks (but not the core clock) and negating the stop mode signal to the bus masters and bus slaves. The only difference is that the CPU will remain in the low power mode with the CPU clock disabled.

During Compute Operation, a DMA 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 the MCG and PMC would then also enter their appropriate modes.

NOTE

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

An enabled DMA 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.

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.

5.2.3 Compute Operation

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

NOTE

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

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

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

Compute Operation is controlled by the CPO register in the MCM (MCM_CPO), which is only accessible to the CPU. Setting or clearing MCM_CPO[CPOREQ] 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 field in the CPO register of MCM module (MCM_CPO[CPOACK]) indicates when entry has completed.

- When exiting Compute Operation in Run mode, MCM_CPO[CPOACK] 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 that MCM_CPO[CPOACK] is polled to determine when the AIPS peripheral space can be accessed without generating a bus error.

The DMA wake-up is also supported during Compute Operation and causes MCM_CPO[CPOACK] to clear and the AIPS peripheral space to be accessible for the duration of the DMA wakeup. At the completion of the DMA wake-up, the device transitions back into Compute Operation.

5.2.4 Peripheral Doze

Several peripherals support a Peripheral Doze mode, where a register bit can be used to disable the peripheral for the duration of a low-power mode. The flash 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.

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.

5.2.5 Clock gating

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

5.3 Power modes

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

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

For each run mode, there is a corresponding Wait and Stop mode. Wait modes are similar to ARM Sleep modes. Stop modes (VLPS, STOP) are similar to ARM Sleep Deep mode. The Very Low Power Run (VLPR) operating mode can drastically reduce runtime power when the maximum bus frequency is not required to handle the application needs.

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

Table 5-1. Chip power modes

Chip mode	Description	Core mode	Normal recovery method
Normal Run	Allows maximum performance of chip. • Default mode out of reset • On-chip voltage regulator is on.	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. • 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. Only MCG modes BLPI and BLPE can be used in VLPR. • Reduced frequency Flash access mode (1 MHz) • LVD off • In BLPI clock mode, only the fast internal reference oscillator is available to provide a low power nominal 4 MHz source for the core with the nominal bus and flash clock required to be <800 kHz • Alternatively, BLPE clock mode can be used with an external clock or the crystal oscillator providing the clock source.	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 (FCLK = 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. • Peripheral clocks are stopped, but OSC, LPTMR, RTC, CMP, TSI can be used. • TPM and UART can optionally be enabled if their clock source is enabled. • NVIC is disabled (FCLK = 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

Table continues on the next page...

Table 5-1. Chip power modes (continued)

Chip mode	Description	Core mode	Normal recovery method
LLS (Low-Leakage Stop)	State retention power mode • Most peripherals are in state retention mode (with clocks stopped), but OSC, LLWU,LPTMR, RTC, CMP, TSI can be used. • NVIC is disabled; LLWU is used to wake up. **NOTE:** The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery • All SRAM is operating (content retained and I/O states held).	Sleep Deep	Wake-up Interrupt[1]
VLLS3 (Very Low-Leakage Stop3)	• Most peripherals are disabled (with clocks stopped), but OSC, LLWU, LPTMR, RTC, CMP, TSI can be used. • NVIC is disabled; LLWU is used to wake up. • SRAM_U and SRAM_L remain powered on (content retained and I/O states held).	Sleep Deep	Wake-up Reset[2]
VLLS1 (Very Low-Leakage Stop1)	• Most peripherals are disabled (with clocks stopped), but OSC, LLWU, LPTMR, RTC, CMP, TSI can be used. • NVIC is disabled; LLWU is used to wake up. • All of SRAM_U and SRAM_L are powered off. The 32-byte system register file remains powered for customer-critical data.	Sleep Deep	Wake-up Reset[2]
VLLS0 (Very Low-Leakage Stop 0)	• Most peripherals are disabled (with clocks stopped), but LLWU, LPTMR, RTC, TSI can be used. • NVIC is disabled; LLWU is used to wake up. • All of SRAM_U and SRAM_L are powered off. The 32-byte system register file remains powered for customer-critical data. • LPO disabled, optional POR brown-out detection	Sleep Deep	Wake-up Reset[2]

1. Resumes Normal Run mode operation by executing the LLWU interrupt service routine.
2. Follows the reset flow with the LLWU interrupt flag set for the NVIC.

5.4 Entering and exiting power modes

The WFI instruction invokes wait and stop modes for the chip. The processor exits the low-power mode via an interrupt.

For LLS and VLLS modes, the wake-up sources are limited to LLWU generated wake-ups, $\overline{\text{NMI\_b}}$ pin, or $\overline{\text{RESET\_b}}$ pin assertions. When the $\overline{\text{NMI\_b}}$ pin or $\overline{\text{RESET\_b}}$ pin have been disabled through associated FTFA_FOPT settings, then these pins are ignored as wakeup sources. The wake-up flow from VLLSx is always through reset.

NOTE

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

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

5.5 Module operation in low-power modes

The table found here illustrates the functionality of each module while the chip is in each of the low power modes.

The standard behavior is shown with some exceptions for Compute Operation (CPO) and Partial Stop2 (PSTOP2).

Debug modules are discussed separately; see Debug in low-power modes. Number ratings (such as 4 MHz and 1 Mbit/s) represent the maximum frequencies or maximum data rates per mode. Following is list of terms also used in the table.

- 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 wake-up. For clocks, OFF means disabled.
- wakeup = Modules can serve as a wake-up source for the chip.

Table 5-2. Module operation in low power modes

Modules	VLPR	VLPW	Stop	VLPS	LLS	VLLSx
Core modules						
NVIC	FF	FF	static	static	static	OFF
System modules						
Mode Controller	FF	FF	FF	FF	FF	FF
LLWU[1]	static	static	static	static	FF	FF[2]
Regulator	low power	low power	ON	low power	low power	low power in VLLS3, OFF in VLLS0/1
LVD	disabled	disabled	ON	disabled	disabled	disabled
Brown-out Detection	ON	ON	ON	ON	ON	ON in VLLS1/3, optionally disabled in VLLS0[3]

Table continues on the next page...

Table 5-2. Module operation in low power modes (continued)

Modules	VLPR	VLPW	Stop	VLPS	LLS	VLLSx
DMA	FF Async operation in CPO	FF	Async operation	Async operation	static	OFF
Watchdog	FF static in CPO	FF	static FF in PSTOP2	static	static	OFF
Clocks						
1kHz LPO	ON	ON	ON	ON	ON	ON in VLLS1/3, OFF in VLLS0
System oscillator (OSC)	OSCERCLK max of 16MHz crystal	OSCERCLK max of 16MHz crystal	OSCERCLK optional	OSCERCLK max of 16MHz crystal	limited to low range/low power	limited to low range/low power in VLLS1/3, OFF in VLLS0
MCG	4 MHz IRC	4 MHz IRC	static - MCGIRCLK optional; PLL optional	static - MCGIRCLK optional	static - no clock output	OFF
Core clock	4 MHz max	OFF	OFF	OFF	OFF	OFF
Platform clock	4 MHz max	4 MHz max	OFF	OFF	OFF	OFF
System clock	4 MHz max OFF in CPO	4 MHz max	OFF	OFF	OFF	OFF
Bus clock	1 MHz max OFF in CPO	1 MHz max	OFF 24 MHz max in PSTOP2 from RUN 1 MHz max in PSTOP2 from VLPR	OFF	OFF	OFF
Memory and memory interfaces						
Flash	1 MHz max access - no program No register access in CPO	low power	low power	low power	OFF	OFF
SRAM_U and SRAM_L	low power	low power	low power	low power	low power	low power in VLLS3, OFF in VLLS0/1
System Register File	powered	powered	powered	powered	powered	powered
Communication interfaces						
UART0	1 Mbit/s Async operation in CPO	1 Mbit/s	Async operation FF in PSTOP2	Async operation	static	OFF
UART1 , UART2	62.5 kbit/s static, wakeup on edge in CPO	62.5 kbit/s	static, wakeup on edge FF in PSTOP2	static, wakeup on edge	static	OFF

Table continues on the next page...

Table 5-2. Module operation in low power modes (continued)

Modules	VLPR	VLPW	Stop	VLPS	LLS	VLLSx
SPI0	master mode 500 kbit/s, slave mode 250 kbit/s static, slave mode receive in CPO	master mode 500 kbit/s, slave mode 250 kbit/s	static, slave mode receive FF in PSTOP2	static, slave mode receive	static	OFF
SPI1	master mode 2 Mbit/s, slave mode 1 Mbit/s static, slave mode receive in CPO	master mode 2 Mbit/s, slave mode 1 Mbit/s	static, slave mode receive	static, slave mode receive	static	OFF
I^2C0	50 kbit/s static, address match wakeup in CPO	50 kbit/s	static, address match wakeup FF in PSTOP2	static, address match wakeup	static	OFF
I^2C1	100 kbps static, address match wakeup in CPO	100 kbps	static, address match wakeup	static, address match wakeup	static	OFF
I^2S	FF Async operation in CPO	FF	Async operation FF in PSTOP2	Async operation	static	OFF
Timers						
TPM	FF Async operation in CPO	FF	Async operation FF in PSTOP2	Async operation	static	OFF
PIT	FF static in CPO	FF	static	static	static	OFF
LPTMR	FF	FF	Async operation FF in PSTOP2	Async operation	Async operation	Async operation[4]
RTC	FF Async operation in CPO	FF	Async operation FF in PSTOP2	Async operation	Async operation	Async operation[5]
Analog						
16-bit ADC	FF ADC internal clock only in CPO	FF	ADC internal clock only FF in PSTOP2	ADC internal clock only	static	OFF
CMP[6]	FF HS or LS compare in CPO	FF	HS or LS compare FF in PSTOP2	HS or LS compare	LS compare	LS compare in VLLS1/3, OFF in VLLS0

Table continues on the next page...

Table 5-2. Module operation in low power modes (continued)

Modules	VLPR	VLPW	Stop	VLPS	LLS	VLLSx
6-bit DAC	FF static in CPO	FF	static FF in PSTOP2	static	static	static, OFF in VLLS0
12-bit DAC	FF static in CPO	FF	static FF in PSTOP2	static	static	static
Human-machine interfaces						
GPIO	FF IOPORT write only in CPO	FF	static output, wakeup input FF in PSTOP2	static output, wakeup input	static, pins latched	OFF, pins latched
TSI	FF Async operation in CPO	Async operation[7]	Async operation[7] FF in PSTOP2	Async operation[7]	Async operation[7]	Async operation[7]

1. Using the LLWU module, the external pins available for this chip do not require the associated peripheral function to be enabled. It only requires the function controlling the pin (GPIO or peripheral) to be configured as an input to allow a transition to occur to the LLWU.
2. Since LPO clock source is disabled, filters will be bypassed during VLLS0.
3. STOPCTRL[PORPO] in the SMC module controls this option.
4. LPO clock source is not available in VLLS0. Also, to use system OSC in VLLS0 it must be configured for bypass (external clock) operation. Pulse counting is available in all modes.
5. In VLLS0 the only clocking option is from RTC_CLKIN.
6. CMP in stop or VLPS supports high speed or low speed external pin to pin or external pin to DAC compares. CMP in LLS or VLLSx only supports low speed external pin to pin or external pin to DAC compares. Windowed, sampled & filtered modes of operation are not available while in Stop, VLPS, LLS, or VLLSx modes.
7. TSI wake-up from all low-power modes is limited to a single selectable pin.

Chapter 6
Debug

6.1 Introduction

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

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

Only one debug interface is supported:

- Serial Wire Debug (SWD)

6.2 Debug port pin descriptions

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

Table 6-1. Serial wire debug pin description

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

6.3 SWD status and control registers

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

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

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

Table 6-2. MDM-AP register summary

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

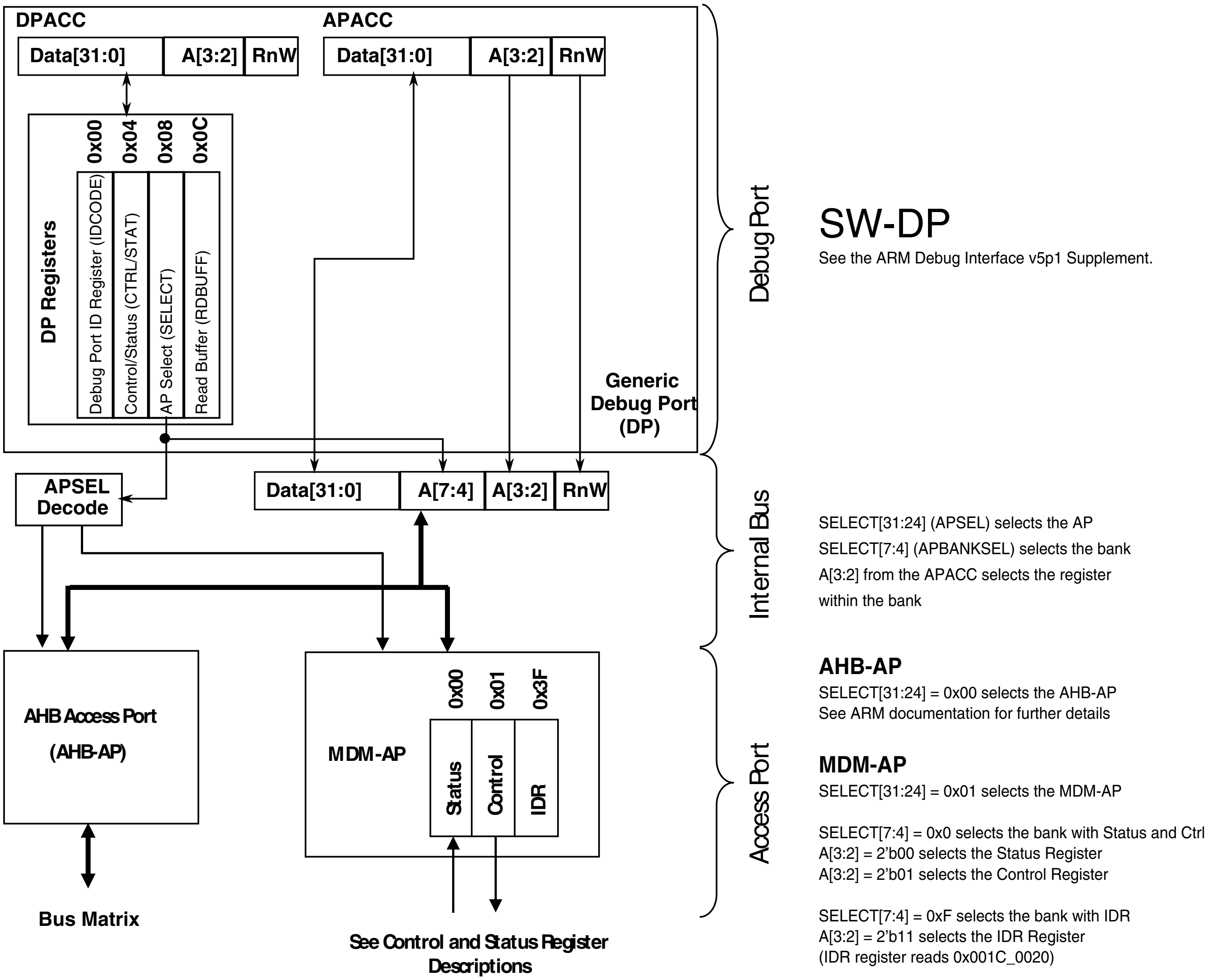


Figure 6-1. MDM AP addressing

6.3.1 MDM-AP Control Register

Table 6-3. MDM-AP Control register assignments

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

Table continues on the next page...

Table 6-3. MDM-AP Control register assignments (continued)

Bit	Name	Secure[1]	Description
2	Debug Request	N	Set to force the core to halt. If the core is in a Stop or Wait mode, this bit can be used to wake the core and transition to a halted state.
3	System Reset Request	Y	Set to force a system reset. The system remains held in reset until this bit is cleared.
4	Core Hold Reset	N	Configuration bit to control core operation at the end of system reset sequencing. 0 Normal operation: Release the core from reset along with the rest of the system at the end of system reset sequencing. 1 Suspend operation: Hold the core in reset at the end of reset sequencing. Once the system enters this suspended state, clearing this control bit immediately releases the core from reset and CPU operation begins.
5	VLLSx Debug Request (VLLDBGREQ)	N	Set to configure the system to be held in reset after the next recovery from a VLLSx mode. This bit is ignored on a VLLS wakeup via the Reset pin. During a VLLS wakeup via the Reset pin, the system can be held in reset by holding the reset pin asserted allowing the debugger to reinitialize the debug modules. This bit holds the system in reset when VLLSx modes are exited to allow the debugger time to re-initialize debug IP before the debug session continues. The Mode Controller captures this bit logic on entry to VLLSx modes. Upon exit from VLLSx modes, the Mode Controller will hold the system in reset until VLLDBGACK is asserted. VLLDBGREQ clears automatically due to the POR reset generated as part of the VLLSx recovery.
6	VLLSx Debug Acknowledge (VLLDBGACK)	N	Set to release a system being held in reset following a VLLSx recovery This bit is used by the debugger to release the system reset when it is being held on VLLSx mode exit. The debugger re-initializes all debug IP and then assert this control bit to allow the Mode Controller to release the system from reset and allow CPU operation to begin. VLLDBGACK is cleared by the debugger or can be left set because it clears automatically due to the POR reset generated as part of the next VLLSx recovery.
7	LLS, VLLSx Status Acknowledge	N	Set this bit to acknowledge the DAP LLS and VLLS Status bits have been read. This acknowledge automatically clears the status bits. This bit is used by the debugger to clear the sticky LLS and VLLSx mode entry status bits. This bit is asserted and cleared by the debugger.
8 – 31	Reserved for future use	N	

1. Command available in secure mode

6.3.2 MDM-AP Status Register

Table 6-4. 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	Indicates Flash has been initialized and debugger can be configured even if system is continuing to be held in reset via the debugger.
2	System Security	Indicates the security state. When secure, the debugger does not have access to the system bus or any memory mapped peripherals. This bit indicates when the part is locked and no system bus access is possible.
3	System Reset	Indicates the system reset state. 0 System is in reset. 1 System is not in reset.
4	Reserved	
5	Mass Erase Enable	Indicates if the MCU can be mass erased or not 0 Mass erase is disabled. 1 Mass erase is enabled .
6	Backdoor Access Key Enable	Indicates if the MCU has the backdoor access key enabled. 0 Disabled 1 Enabled
7	LP Enabled	Decode of SMC_PMCTRL[STOPM] field to indicate that VLPS, LLS, or VLLSx are the selected power mode the next time the ARM Core enters Deep Sleep. 0 Low Power Stop Mode is not enabled. 1 Low Power Stop Mode is enabled. Usage intended for debug operation in which Run to VLPS is attempted. Per debug definition, the system actually enters the Stop state. A debugger should interpret deep sleep indication (with SLEEPDEEP and SLEEPING asserted), in conjunction with this bit asserted as the debugger-VLPS status indication.
8	Very Low Power Mode	Indicates current power mode is VLPx. This bit is not ‘sticky’ and should always represent whether VLPx is enabled or not. This bit is used to throttle SWD_CLK frequency up/down.

Table continues on the next page...

Table 6-4. MDM-AP Status register assignments (continued)

Bit	Name	Description
9	LLS Mode Exit	This bit indicates an exit from LLS mode has occurred. The debugger will lose communication while the system is in LLS (including access to this register). Once communication is reestablished, this bit indicates that the system had been in LLS. Since the debug modules held their state during LLS, they do not need to be reconfigured. This bit is set during the LLS recovery sequence. The LLS Mode Exit bit is held until the debugger has had a chance to recognize that LLS was exited and is cleared by a write of 1 to the LLS, VLLSx Status Acknowledge bit in MDM AP Control register.
10	VLLSx Modes Exit	This bit indicates an exit from VLLSx mode has occurred. The debugger will lose communication while the system is in VLLSx (including access to this register). Once communication is reestablished, this bit indicates that the system had been in VLLSx. Since the debug modules lose their state during VLLSx modes, they need to be reconfigured. This bit is set during the VLLSx recovery sequence. The VLLSx Mode Exit bit is held until the debugger has had a chance to recognize that a VLLS mode was exited and is cleared by a write of 1 to the LLS, VLLSx Status Acknowledge bit in MDM AP Control register.
11 – 15	Reserved for future use	Always read 0.
16	Core Halted	Indicates the core has entered Debug Halt mode
17	Core SLEEPDEEP	Indicates the core has entered a low-power mode SLEEPING==1 and SLEEPDEEP==0 indicates wait or VLPW mode. SLEEPING==1 and SLEEPDEEP==1 indicates stop or VLPS mode.
18	Core SLEEPING	
19 – 31	Reserved for future use	Always read 0.

6.4 Debug resets

The debug system receives the following sources of reset:

- Debug reset (the CDBGRSTREQ field within the DP CTRL/STAT register) 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:

- A system reset in the DAP control register which allows the debugger to hold the system in reset.
- SYSRESETREQ field in the NVIC Application Interrupt and Reset control register
- A system reset in the DAP control register which allows the debugger to hold the core in reset.

6.5 Micro Trace Buffer (MTB)

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

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

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

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

6.6 Debug in low-power modes

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

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

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

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

NOTE

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

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

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

6.7 Debug and security

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

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

Chapter 7
Port Control and Interrupts (PORT)

7.1 Introduction

NOTE

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

7.2 Overview

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

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

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

7.2.1 Features

The PORT module has the following features:

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

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

7.2.2 Modes of operation

7.2.2.1 Run mode

In Run mode, the PORT operates normally.

7.2.2.2 Wait mode

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

7.2.2.3 Stop mode

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

7.2.2.4 Debug mode

In Debug mode, PORT operates normally.

7.3 External signal description

The table found here describes the PORT external signal.

Table 7-1. 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.

7.4 Detailed signal description

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

Table 7-2. 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.

7.5 Memory map and register definition

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

PORT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_9000	Pin Control Register n (PORTA_PCR0)	32	R/W	See section	7.5.1/141
4004_9004	Pin Control Register n (PORTA_PCR1)	32	R/W	See section	7.5.1/141
4004_9008	Pin Control Register n (PORTA_PCR2)	32	R/W	See section	7.5.1/141

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_900C	Pin Control Register n (PORTA_PCR3)	32	R/W	See section	7.5.1/141
4004_9010	Pin Control Register n (PORTA_PCR4)	32	R/W	See section	7.5.1/141
4004_9014	Pin Control Register n (PORTA_PCR5)	32	R/W	See section	7.5.1/141
4004_9018	Pin Control Register n (PORTA_PCR6)	32	R/W	See section	7.5.1/141
4004_901C	Pin Control Register n (PORTA_PCR7)	32	R/W	See section	7.5.1/141
4004_9020	Pin Control Register n (PORTA_PCR8)	32	R/W	See section	7.5.1/141
4004_9024	Pin Control Register n (PORTA_PCR9)	32	R/W	See section	7.5.1/141
4004_9028	Pin Control Register n (PORTA_PCR10)	32	R/W	See section	7.5.1/141
4004_902C	Pin Control Register n (PORTA_PCR11)	32	R/W	See section	7.5.1/141
4004_9030	Pin Control Register n (PORTA_PCR12)	32	R/W	See section	7.5.1/141
4004_9034	Pin Control Register n (PORTA_PCR13)	32	R/W	See section	7.5.1/141
4004_9038	Pin Control Register n (PORTA_PCR14)	32	R/W	See section	7.5.1/141
4004_903C	Pin Control Register n (PORTA_PCR15)	32	R/W	See section	7.5.1/141
4004_9040	Pin Control Register n (PORTA_PCR16)	32	R/W	See section	7.5.1/141
4004_9044	Pin Control Register n (PORTA_PCR17)	32	R/W	See section	7.5.1/141
4004_9048	Pin Control Register n (PORTA_PCR18)	32	R/W	See section	7.5.1/141
4004_904C	Pin Control Register n (PORTA_PCR19)	32	R/W	See section	7.5.1/141
4004_9050	Pin Control Register n (PORTA_PCR20)	32	R/W	See section	7.5.1/141
4004_9054	Pin Control Register n (PORTA_PCR21)	32	R/W	See section	7.5.1/141
4004_9058	Pin Control Register n (PORTA_PCR22)	32	R/W	See section	7.5.1/141
4004_905C	Pin Control Register n (PORTA_PCR23)	32	R/W	See section	7.5.1/141
4004_9060	Pin Control Register n (PORTA_PCR24)	32	R/W	See section	7.5.1/141
4004_9064	Pin Control Register n (PORTA_PCR25)	32	R/W	See section	7.5.1/141
4004_9068	Pin Control Register n (PORTA_PCR26)	32	R/W	See section	7.5.1/141
4004_906C	Pin Control Register n (PORTA_PCR27)	32	R/W	See section	7.5.1/141
4004_9070	Pin Control Register n (PORTA_PCR28)	32	R/W	See section	7.5.1/141
4004_9074	Pin Control Register n (PORTA_PCR29)	32	R/W	See section	7.5.1/141
4004_9078	Pin Control Register n (PORTA_PCR30)	32	R/W	See section	7.5.1/141
4004_907C	Pin Control Register n (PORTA_PCR31)	32	R/W	See section	7.5.1/141
4004_9080	Global Pin Control Low Register (PORTA_GPCLR)	32	W (always reads 0)	0000_0000h	7.5.2/143
4004_9084	Global Pin Control High Register (PORTA_GPCHR)	32	W (always reads 0)	0000_0000h	7.5.3/144
4004_90A0	Interrupt Status Flag Register (PORTA_ISFR)	32	w1c	0000_0000h	7.5.4/144
4004_A000	Pin Control Register n (PORTB_PCR0)	32	R/W	See section	7.5.1/141
4004_A004	Pin Control Register n (PORTB_PCR1)	32	R/W	See section	7.5.1/141
4004_A008	Pin Control Register n (PORTB_PCR2)	32	R/W	See section	7.5.1/141

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_A00C	Pin Control Register n (PORTB_PCR3)	32	R/W	See section	7.5.1/141
4004_A010	Pin Control Register n (PORTB_PCR4)	32	R/W	See section	7.5.1/141
4004_A014	Pin Control Register n (PORTB_PCR5)	32	R/W	See section	7.5.1/141
4004_A018	Pin Control Register n (PORTB_PCR6)	32	R/W	See section	7.5.1/141
4004_A01C	Pin Control Register n (PORTB_PCR7)	32	R/W	See section	7.5.1/141
4004_A020	Pin Control Register n (PORTB_PCR8)	32	R/W	See section	7.5.1/141
4004_A024	Pin Control Register n (PORTB_PCR9)	32	R/W	See section	7.5.1/141
4004_A028	Pin Control Register n (PORTB_PCR10)	32	R/W	See section	7.5.1/141
4004_A02C	Pin Control Register n (PORTB_PCR11)	32	R/W	See section	7.5.1/141
4004_A030	Pin Control Register n (PORTB_PCR12)	32	R/W	See section	7.5.1/141
4004_A034	Pin Control Register n (PORTB_PCR13)	32	R/W	See section	7.5.1/141
4004_A038	Pin Control Register n (PORTB_PCR14)	32	R/W	See section	7.5.1/141
4004_A03C	Pin Control Register n (PORTB_PCR15)	32	R/W	See section	7.5.1/141
4004_A040	Pin Control Register n (PORTB_PCR16)	32	R/W	See section	7.5.1/141
4004_A044	Pin Control Register n (PORTB_PCR17)	32	R/W	See section	7.5.1/141
4004_A048	Pin Control Register n (PORTB_PCR18)	32	R/W	See section	7.5.1/141
4004_A04C	Pin Control Register n (PORTB_PCR19)	32	R/W	See section	7.5.1/141
4004_A050	Pin Control Register n (PORTB_PCR20)	32	R/W	See section	7.5.1/141
4004_A054	Pin Control Register n (PORTB_PCR21)	32	R/W	See section	7.5.1/141
4004_A058	Pin Control Register n (PORTB_PCR22)	32	R/W	See section	7.5.1/141
4004_A05C	Pin Control Register n (PORTB_PCR23)	32	R/W	See section	7.5.1/141
4004_A060	Pin Control Register n (PORTB_PCR24)	32	R/W	See section	7.5.1/141
4004_A064	Pin Control Register n (PORTB_PCR25)	32	R/W	See section	7.5.1/141
4004_A068	Pin Control Register n (PORTB_PCR26)	32	R/W	See section	7.5.1/141
4004_A06C	Pin Control Register n (PORTB_PCR27)	32	R/W	See section	7.5.1/141
4004_A070	Pin Control Register n (PORTB_PCR28)	32	R/W	See section	7.5.1/141
4004_A074	Pin Control Register n (PORTB_PCR29)	32	R/W	See section	7.5.1/141
4004_A078	Pin Control Register n (PORTB_PCR30)	32	R/W	See section	7.5.1/141
4004_A07C	Pin Control Register n (PORTB_PCR31)	32	R/W	See section	7.5.1/141
4004_A080	Global Pin Control Low Register (PORTB_GPCLR)	32	W (always reads 0)	0000_0000h	7.5.2/143
4004_A084	Global Pin Control High Register (PORTB_GPCHR)	32	W (always reads 0)	0000_0000h	7.5.3/144
4004_A0A0	Interrupt Status Flag Register (PORTB_ISFR)	32	w1c	0000_0000h	7.5.4/144
4004_B000	Pin Control Register n (PORTC_PCR0)	32	R/W	See section	7.5.1/141
4004_B004	Pin Control Register n (PORTC_PCR1)	32	R/W	See section	7.5.1/141
4004_B008	Pin Control Register n (PORTC_PCR2)	32	R/W	See section	7.5.1/141

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_B00C	Pin Control Register n (PORTC_PCR3)	32	R/W	See section	7.5.1/141
4004_B010	Pin Control Register n (PORTC_PCR4)	32	R/W	See section	7.5.1/141
4004_B014	Pin Control Register n (PORTC_PCR5)	32	R/W	See section	7.5.1/141
4004_B018	Pin Control Register n (PORTC_PCR6)	32	R/W	See section	7.5.1/141
4004_B01C	Pin Control Register n (PORTC_PCR7)	32	R/W	See section	7.5.1/141
4004_B020	Pin Control Register n (PORTC_PCR8)	32	R/W	See section	7.5.1/141
4004_B024	Pin Control Register n (PORTC_PCR9)	32	R/W	See section	7.5.1/141
4004_B028	Pin Control Register n (PORTC_PCR10)	32	R/W	See section	7.5.1/141
4004_B02C	Pin Control Register n (PORTC_PCR11)	32	R/W	See section	7.5.1/141
4004_B030	Pin Control Register n (PORTC_PCR12)	32	R/W	See section	7.5.1/141
4004_B034	Pin Control Register n (PORTC_PCR13)	32	R/W	See section	7.5.1/141
4004_B038	Pin Control Register n (PORTC_PCR14)	32	R/W	See section	7.5.1/141
4004_B03C	Pin Control Register n (PORTC_PCR15)	32	R/W	See section	7.5.1/141
4004_B040	Pin Control Register n (PORTC_PCR16)	32	R/W	See section	7.5.1/141
4004_B044	Pin Control Register n (PORTC_PCR17)	32	R/W	See section	7.5.1/141
4004_B048	Pin Control Register n (PORTC_PCR18)	32	R/W	See section	7.5.1/141
4004_B04C	Pin Control Register n (PORTC_PCR19)	32	R/W	See section	7.5.1/141
4004_B050	Pin Control Register n (PORTC_PCR20)	32	R/W	See section	7.5.1/141
4004_B054	Pin Control Register n (PORTC_PCR21)	32	R/W	See section	7.5.1/141
4004_B058	Pin Control Register n (PORTC_PCR22)	32	R/W	See section	7.5.1/141
4004_B05C	Pin Control Register n (PORTC_PCR23)	32	R/W	See section	7.5.1/141
4004_B060	Pin Control Register n (PORTC_PCR24)	32	R/W	See section	7.5.1/141
4004_B064	Pin Control Register n (PORTC_PCR25)	32	R/W	See section	7.5.1/141
4004_B068	Pin Control Register n (PORTC_PCR26)	32	R/W	See section	7.5.1/141
4004_B06C	Pin Control Register n (PORTC_PCR27)	32	R/W	See section	7.5.1/141
4004_B070	Pin Control Register n (PORTC_PCR28)	32	R/W	See section	7.5.1/141
4004_B074	Pin Control Register n (PORTC_PCR29)	32	R/W	See section	7.5.1/141
4004_B078	Pin Control Register n (PORTC_PCR30)	32	R/W	See section	7.5.1/141
4004_B07C	Pin Control Register n (PORTC_PCR31)	32	R/W	See section	7.5.1/141
4004_B080	Global Pin Control Low Register (PORTC_GPCLR)	32	W (always reads 0)	0000_0000h	7.5.2/143
4004_B084	Global Pin Control High Register (PORTC_GPCHR)	32	W (always reads 0)	0000_0000h	7.5.3/144
4004_B0A0	Interrupt Status Flag Register (PORTC_ISFR)	32	w1c	0000_0000h	7.5.4/144
4004_C000	Pin Control Register n (PORTD_PCR0)	32	R/W	See section	7.5.1/141
4004_C004	Pin Control Register n (PORTD_PCR1)	32	R/W	See section	7.5.1/141
4004_C008	Pin Control Register n (PORTD_PCR2)	32	R/W	See section	7.5.1/141

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_C00C	Pin Control Register n (PORTD_PCR3)	32	R/W	See section	7.5.1/141
4004_C010	Pin Control Register n (PORTD_PCR4)	32	R/W	See section	7.5.1/141
4004_C014	Pin Control Register n (PORTD_PCR5)	32	R/W	See section	7.5.1/141
4004_C018	Pin Control Register n (PORTD_PCR6)	32	R/W	See section	7.5.1/141
4004_C01C	Pin Control Register n (PORTD_PCR7)	32	R/W	See section	7.5.1/141
4004_C020	Pin Control Register n (PORTD_PCR8)	32	R/W	See section	7.5.1/141
4004_C024	Pin Control Register n (PORTD_PCR9)	32	R/W	See section	7.5.1/141
4004_C028	Pin Control Register n (PORTD_PCR10)	32	R/W	See section	7.5.1/141
4004_C02C	Pin Control Register n (PORTD_PCR11)	32	R/W	See section	7.5.1/141
4004_C030	Pin Control Register n (PORTD_PCR12)	32	R/W	See section	7.5.1/141
4004_C034	Pin Control Register n (PORTD_PCR13)	32	R/W	See section	7.5.1/141
4004_C038	Pin Control Register n (PORTD_PCR14)	32	R/W	See section	7.5.1/141
4004_C03C	Pin Control Register n (PORTD_PCR15)	32	R/W	See section	7.5.1/141
4004_C040	Pin Control Register n (PORTD_PCR16)	32	R/W	See section	7.5.1/141
4004_C044	Pin Control Register n (PORTD_PCR17)	32	R/W	See section	7.5.1/141
4004_C048	Pin Control Register n (PORTD_PCR18)	32	R/W	See section	7.5.1/141
4004_C04C	Pin Control Register n (PORTD_PCR19)	32	R/W	See section	7.5.1/141
4004_C050	Pin Control Register n (PORTD_PCR20)	32	R/W	See section	7.5.1/141
4004_C054	Pin Control Register n (PORTD_PCR21)	32	R/W	See section	7.5.1/141
4004_C058	Pin Control Register n (PORTD_PCR22)	32	R/W	See section	7.5.1/141
4004_C05C	Pin Control Register n (PORTD_PCR23)	32	R/W	See section	7.5.1/141
4004_C060	Pin Control Register n (PORTD_PCR24)	32	R/W	See section	7.5.1/141
4004_C064	Pin Control Register n (PORTD_PCR25)	32	R/W	See section	7.5.1/141
4004_C068	Pin Control Register n (PORTD_PCR26)	32	R/W	See section	7.5.1/141
4004_C06C	Pin Control Register n (PORTD_PCR27)	32	R/W	See section	7.5.1/141
4004_C070	Pin Control Register n (PORTD_PCR28)	32	R/W	See section	7.5.1/141
4004_C074	Pin Control Register n (PORTD_PCR29)	32	R/W	See section	7.5.1/141
4004_C078	Pin Control Register n (PORTD_PCR30)	32	R/W	See section	7.5.1/141
4004_C07C	Pin Control Register n (PORTD_PCR31)	32	R/W	See section	7.5.1/141
4004_C080	Global Pin Control Low Register (PORTD_GPCLR)	32	W (always reads 0)	0000_0000h	7.5.2/143
4004_C084	Global Pin Control High Register (PORTD_GPCHR)	32	W (always reads 0)	0000_0000h	7.5.3/144
4004_C0A0	Interrupt Status Flag Register (PORTD_ISFR)	32	w1c	0000_0000h	7.5.4/144
4004_D000	Pin Control Register n (PORTE_PCR0)	32	R/W	See section	7.5.1/141
4004_D004	Pin Control Register n (PORTE_PCR1)	32	R/W	See section	7.5.1/141
4004_D008	Pin Control Register n (PORTE_PCR2)	32	R/W	See section	7.5.1/141

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_D00C	Pin Control Register n (PORTE_PCR3)	32	R/W	See section	7.5.1/141
4004_D010	Pin Control Register n (PORTE_PCR4)	32	R/W	See section	7.5.1/141
4004_D014	Pin Control Register n (PORTE_PCR5)	32	R/W	See section	7.5.1/141
4004_D018	Pin Control Register n (PORTE_PCR6)	32	R/W	See section	7.5.1/141
4004_D01C	Pin Control Register n (PORTE_PCR7)	32	R/W	See section	7.5.1/141
4004_D020	Pin Control Register n (PORTE_PCR8)	32	R/W	See section	7.5.1/141
4004_D024	Pin Control Register n (PORTE_PCR9)	32	R/W	See section	7.5.1/141
4004_D028	Pin Control Register n (PORTE_PCR10)	32	R/W	See section	7.5.1/141
4004_D02C	Pin Control Register n (PORTE_PCR11)	32	R/W	See section	7.5.1/141
4004_D030	Pin Control Register n (PORTE_PCR12)	32	R/W	See section	7.5.1/141
4004_D034	Pin Control Register n (PORTE_PCR13)	32	R/W	See section	7.5.1/141
4004_D038	Pin Control Register n (PORTE_PCR14)	32	R/W	See section	7.5.1/141
4004_D03C	Pin Control Register n (PORTE_PCR15)	32	R/W	See section	7.5.1/141
4004_D040	Pin Control Register n (PORTE_PCR16)	32	R/W	See section	7.5.1/141
4004_D044	Pin Control Register n (PORTE_PCR17)	32	R/W	See section	7.5.1/141
4004_D048	Pin Control Register n (PORTE_PCR18)	32	R/W	See section	7.5.1/141
4004_D04C	Pin Control Register n (PORTE_PCR19)	32	R/W	See section	7.5.1/141
4004_D050	Pin Control Register n (PORTE_PCR20)	32	R/W	See section	7.5.1/141
4004_D054	Pin Control Register n (PORTE_PCR21)	32	R/W	See section	7.5.1/141
4004_D058	Pin Control Register n (PORTE_PCR22)	32	R/W	See section	7.5.1/141
4004_D05C	Pin Control Register n (PORTE_PCR23)	32	R/W	See section	7.5.1/141
4004_D060	Pin Control Register n (PORTE_PCR24)	32	R/W	See section	7.5.1/141
4004_D064	Pin Control Register n (PORTE_PCR25)	32	R/W	See section	7.5.1/141
4004_D068	Pin Control Register n (PORTE_PCR26)	32	R/W	See section	7.5.1/141
4004_D06C	Pin Control Register n (PORTE_PCR27)	32	R/W	See section	7.5.1/141
4004_D070	Pin Control Register n (PORTE_PCR28)	32	R/W	See section	7.5.1/141
4004_D074	Pin Control Register n (PORTE_PCR29)	32	R/W	See section	7.5.1/141
4004_D078	Pin Control Register n (PORTE_PCR30)	32	R/W	See section	7.5.1/141
4004_D07C	Pin Control Register n (PORTE_PCR31)	32	R/W	See section	7.5.1/141
4004_D080	Global Pin Control Low Register (PORTE_GPCLR)	32	W (always reads 0)	0000_0000h	7.5.2/143
4004_D084	Global Pin Control High Register (PORTE_GPCHR)	32	W (always reads 0)	0000_0000h	7.5.3/144
4004_D0A0	Interrupt Status Flag Register (PORTE_ISFR)	32	w1c	0000_0000h	7.5.4/144

7.5.1 Pin Control Register n (PORTx_PCRn)

NOTE

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

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

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

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

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

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

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

PORTx_PCR*n* field descriptions

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

Table continues on the next page...

PORTx_PCRn field descriptions (continued)

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

Table continues on the next page...

PORTx_PCRn field descriptions (continued)

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

7.5.2 Global Pin Control Low Register (PORTx_GPCLR)

Only 32-bit writes are supported to this register.

Address: Base address + 80h offset

Bit	31–16	15–0
R	0	0
W	GPWE	GPWD
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PORTx_GPCLR field descriptions

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

7.5.3 Global Pin Control High Register (PORTx_GPCHR)

Only 32-bit writes are supported to this register.

Address: Base address + 84h offset

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

PORTx_GPCHR field descriptions

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

7.5.4 Interrupt Status Flag Register (PORTx_ISFR)

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

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

Address: Base address + A0h offset

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

PORTx_ISFR field descriptions

Field	Description
31–0 ISF	Interrupt Status Flag Each bit in the field indicates the detection of the configured interrupt of the same number as the field. 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.

7.6 Functional description

7.6.1 Pin control

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

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

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

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

The functions apply across all digital pin muxing modes and individual peripherals do not override the configuration in the Pin Control register. For example, if an I^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, and passive filter enable.

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

7.6.2 Global pin control

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

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

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

7.6.3 External interrupts

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

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

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

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

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

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

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

Chapter 8
System Integration Module (SIM)

8.1 Introduction

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

8.1.1 Features

- System clocking configuration
    - System clock divide values
    - Architectural clock gating control
    - ERCLK32K clock selection
    - UART0 and TPM clock selection
- Flash and System RAM size configuration
- TPM external clock and input capture selection
- UART receive/transmit source selection/configuration

8.2 Memory map and register definition

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

NOTE

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

NOTE

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

SIM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_7000	System Options Register 1 (SIM_SOPT1)	32	R/W	0000_0000h	8.2.1/151
4004_8004	System Options Register 2 (SIM_SOPT2)	32	R/W	0000_0000h	8.2.2/152
4004_800C	System Options Register 4 (SIM_SOPT4)	32	R/W	0000_0000h	8.2.3/154
4004_8010	System Options Register 5 (SIM_SOPT5)	32	R/W	0000_0000h	8.2.4/155
4004_8018	System Options Register 7 (SIM_SOPT7)	32	R/W	0000_0000h	8.2.5/157
4004_8024	System Device Identification Register (SIM_SDID)	32	R		8.2.6/158
4004_8034	System Clock Gating Control Register 4 (SIM_SCGC4)	32	R/W	F000_0030h	8.2.7/160
4004_8038	System Clock Gating Control Register 5 (SIM_SCGC5)	32	R/W	0000_0182h	8.2.8/162
4004_803C	System Clock Gating Control Register 6 (SIM_SCGC6)	32	R/W	0000_0001h	8.2.9/164
4004_8040	System Clock Gating Control Register 7 (SIM_SCGC7)	32	R/W	0000_0100h	8.2.10/166
4004_8044	System Clock Divider Register 1 (SIM_CLKDIV1)	32	R/W	See section	8.2.11/166
4004_804C	Flash Configuration Register 1 (SIM_FCFG1)	32	R/W	See section	8.2.12/168
4004_8050	Flash Configuration Register 2 (SIM_FCFG2)	32	R	See section	8.2.13/169
4004_8058	Unique Identification Register Mid-High (SIM_UIDMH)	32	R	See section	8.2.14/170
4004_805C	Unique Identification Register Mid Low (SIM_UIDML)	32	R	See section	8.2.15/171
4004_8060	Unique Identification Register Low (SIM_UIDL)	32	R	See section	8.2.16/171
4004_8100	COP Control Register (SIM_COPC)	32	R/W	0000_000Ch	8.2.17/172
4004_8104	Service COP (SIM_SRVCOP)	32	W	0000_0000h	8.2.18/173

8.2.1 System Options Register 1 (SIM_SOPT1)

NOTE

The SOPT1 register is only reset on POR or LVD.

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

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

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

SIM_SOPT1 field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–18 OSC32KSEL	32K Oscillator Clock Select Selects the 32 kHz clock source (ERCLK32K) for RTC and LPTMR. This field is reset only on POR/LVD. 00 System oscillator (OSC32KCLK) 01 Reserved 10 RTC_CLKIN 11 LPO 1kHz
17–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

8.2.2 System Options Register 2 (SIM_SOPT2)

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

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0		UART0SRC		TPMSRC		0		0				0	PLLFLLSEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

SIM_SOPT2 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–26 UART0SRC	UART0 Clock Source Select Selects the clock source for the UART0 transmit and receive clock. 00 Clock disabled 01 MCGFLLCLK clock, or MCGPLLCLK/2 10 OSCERCLK clock 11 MCGIRCLK clock
25–24 TPMSRC	TPM Clock Source Select Selects the clock source for the TPM counter clock 00 Clock disabled 01 MCGFLLCLK clock, or MCGPLLCLK/2 10 OSCERCLK clock 11 MCGIRCLK clock

Table continues on the next page...

SIM_SOPT2 field descriptions (continued)

Field	Description
23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 PLLFLLSEL	PLL/FLL clock select Selects the MCGPLLCLK or MCGFLLCLK clock for various peripheral clocking options. 0 MCGFLLCLK clock 1 MCGPLLCLK clock with fixed divide by 2
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 CLKOUTSEL	CLKOUT select Selects the clock to output on the CLKOUT pin. 000 Reserved 001 Reserved 010 Bus clock 011 LPO clock (1 kHz) 100 MCGIRCLK 101 Reserved 110 OSCERCLK 111 Reserved
4 RTCCLKOUTSEL	RTC Clock Out Select Selects either the RTC 1 Hz clock or the OSC clock to be output on the RTC_CLKOUT pin. 0 RTC 1 Hz clock is output on the RTC_CLKOUT pin. 1 OSCERCLK clock is output on the RTC_CLKOUT pin.
3–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

8.2.3 System Options Register 4 (SIM_SOPT4)

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

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

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

SIM_SOPT4 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 TPM2CLKSEL	TPM2 External Clock Pin Select Selects the external pin used to drive the clock to the TPM2 module. **NOTE:** The selected pin must also be configured for the TPM external clock function through the appropriate Pin Control Register in the Port Control module. 0 TPM2 external clock driven by TPM_CLKIN0 pin. 1 TPM2 external clock driven by TPM_CLKIN1 pin.
25 TPM1CLKSEL	TPM1 External Clock Pin Select Selects the external pin used to drive the clock to the TPM1 module. **NOTE:** The selected pin must also be configured for the TPM external clock function through the appropriate pin control register in the port control module. 0 TPM1 external clock driven by TPM_CLKIN0 pin. 1 TPM1 external clock driven by TPM_CLKIN1 pin.
24 TPM0CLKSEL	TPM0 External Clock Pin Select Selects the external pin used to drive the clock to the TPM0 module. **NOTE:** The selected pin must also be configured for the TPM external clock function through the appropriate pin control register in the port control module. 0 TPM0 external clock driven by TPM_CLKIN0 pin. 1 TPM0 external clock driven by TPM_CLKIN1 pin.

Table continues on the next page...

SIM_SOPT4 field descriptions (continued)

Field	Description
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 TPM2CH0SRC	TPM2 Channel 0 Input Capture Source Select Selects the source for TPM2 channel 0 input capture. **NOTE:** When TPM2 is not in input capture mode, clear this field. 0 TPM2_CH0 signal 1 CMP0 output
19–18 TPM1CH0SRC	TPM1 channel 0 input capture source select Selects the source for TPM1 channel 0 input capture. **NOTE:** When TPM1 is not in input capture mode, clear this field. 00 TPM1_CH0 signal 01 CMP0 output 10 Reserved 11 Reserved
17–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

8.2.4 System Options Register 5 (SIM_SOPT5)

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												0	UART2ODE	UART1ODE	UART0ODE
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									UART1RXSRC	UART1TXSRC		0	UART0RXSRC	UART0TXSRC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT5 field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 UART2ODE	UART2 Open Drain Enable 0 Open drain is disabled on UART2 1 Open drain is enabled on UART2
17 UART1ODE	UART1 Open Drain Enable 0 Open drain is disabled on UART1. 1 Open drain is enabled on UART1
16 UART0ODE	UART0 Open Drain Enable 0 Open drain is disabled on UART0. 1 Open drain is enabled on UART0.
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 UART1RXSRC	UART1 Receive Data Source Select Selects the source for the UART1 receive data. 0 UART1_RX pin 1 CMP0 output
5–4 UART1TXSRC	UART1 Transmit Data Source Select Selects the source for the UART1 transmit data. 00 UART1_TX pin 01 UART1_TX pin modulated with TPM1 channel 0 output 10 UART1_TX pin modulated with TPM2 channel 0 output 11 Reserved
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 UART0RXSRC	UART0 Receive Data Source Select Selects the source for the UART0 receive data. 0 UART_RX pin 1 CMP0 output
1–0 UART0TXSRC	UART0 Transmit Data Source Select Selects the source for the UART0 transmit data. 00 UART0_TX pin 01 UART0_TX pin modulated with TPM1 channel 0 output 10 UART0_TX pin modulated with TPM2 channel 0 output 11 Reserved

8.2.5 System Options Register 7 (SIM_SOPT7)

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

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

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

SIM_SOPT7 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADC0ALTTRGEN	ADC0 Alternate Trigger Enable Enables alternative conversion triggers for ADC0. 0 ADC ADHWT trigger comes from TPM1 channel 0 and channel1. Prior to the assertion of TPM1 channel 0, a pre-trigger pulse will be sent to ADHWTSA to initiate an ADC acquisition using ADCx_SC1A configuration and store ADC conversion in ADCx_RA Register. Prior to the assertion of TPM1 channel 1 a pre-trigger pulse will be sent to ADHWTSB to initiate an ADC acquisition using ADCx_SC1Bconfiguration and store ADC conversion in ADCx_RB Register. 1 ADC ADHWT trigger comes from a peripheral event selected by ADC0TRGSEL bits. ADC0PRETRGSEL bit will select the optional ADHWTSA or ADHWTSB select lines for choosing the ADCx_SC1x config and ADCx_Rx result regsiter to store the ADC conversion.
6–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 ADC0PRETRGSEL	ADC0 Pretrigger Select Selects the ADC0 pre-trigger source when alternative triggers are enabled through ADC0ALTTRGEN. **NOTE:** The ADC0PRETRGSEL function is ignored if ADC0ALTTRGEN = 0. 0 Pre-trigger ADHDWTSA is selected, thus ADC0 will use ADC0_SC1A configuration for the next ADC conversion and store the result in ADC0_RA register. 1 Pre-trigger ADHDWTSB is selected, thus ADC0 will use ADC0_SC1B configuration for the next ADC conversion and store the result in ADC0_RB register.

Table continues on the next page...

SIM_SOPT7 field descriptions (continued)

Field	Description
3–0 ADC0TRGSEL	ADC0 Trigger Select Selects 1 of 16 peripherals to initiate an ADC conversion via the ADHWDT input, when ADC0ALTTRGEN =1, else is ignored by ADC0. 0000 External trigger pin input (EXTRG_IN) 0001 CMP0 output 0010 Reserved 0011 Reserved 0100 PIT trigger 0 0101 PIT trigger 1 0110 Reserved 0111 Reserved 1000 TPM0 overflow 1001 TPM1 overflow 1010 TPM2 overflow 1011 Reserved 1100 RTC alarm 1101 RTC seconds 1110 LPTMR0 trigger 1111 Reserved

8.2.6 System Device Identification Register (SIM_SDID)

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

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

* Notes:
- FAMID field: Device specific value.
- SUBFAMID field: Device specific value.
- SRAMSIZE field: Device specific value.
- REVID field: Device specific value.
- PINID field: Device specific value.

SIM_SDID field descriptions

Field	Description
31–28 FAMID	Kinetis family ID Specifies the Kinetis family of the device. 0000 KL0x Family (low end) 0001 KL1x Family (basic)

Table continues on the next page...

SIM_SDID field descriptions (continued)

Field	Description
	0010 KL2x Family (USB) 0011 KL3x Family (Segment LCD) 0100 KL4x Family (USB and Segment LCD)
27–24 SUBFAMID	Kinetis Sub-Family ID Specifies the Kinetis sub-family of the device. 0010 KLx2 Subfamily 0011 KLx3 Subfamily 0100 KLx4 Subfamily 0101 KLx5 Subfamily 0110 KLx6 Subfamily 0111 KLx7 Subfamily
23–20 SERIESID	Kinetis Series ID Specifies the Kinetis family of the device. 0001 KL family
19–16 SRAMSIZE	System SRAM Size Specifies the size of the System SRAM 0000 0.5 KB 0001 1 KB 0010 2 KB 0011 4 KB 0100 8 KB 0101 16 KB 0110 32 KB 0111 64 KB
15–12 REVID	Device Revision Number Specifies the silicon implementation number for the device.
11–7 DIEID	Device Die Number Specifies the silicon implementation number for the device.
6–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 PINID	Pincount Identification Specifies the pincount of the device. 0000 16-pin 0001 24-pin 0010 32-pin 0011 36-pin 0100 48-pin 0101 64-pin 0110 80-pin 0111 Reserved

Table continues on the next page...

SIM_SDID field descriptions (continued)

Field	Description
	1000 100-pin 1001 Reserved 1010 Reserved 1011 Custom pinout (WLCSP) 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved

8.2.7 System Clock Gating Control Register 4 (SIM_SCGC4)

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1				0				SPI1	SPI0	0	0	CMP	0	0	
W																
Reset	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	UART2	UART1	UART0	0	0	I2C1	I2C0	1		0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

SIM_SCGC4 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
27–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 SPI1	SPI1 Clock Gate Control Controls the clock gate to the SPI1 module. 0 Clock disabled 1 Clock enabled
22 SPI0	SPI0 Clock Gate Control Controls the clock gate to the SPI0 module. 0 Clock disabled 1 Clock enabled

Table continues on the next page...

SIM_SCGC4 field descriptions (continued)

Field	Description
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 CMP	Comparator Clock Gate Control Controls the clock gate to the comparator module. 0 Clock disabled 1 Clock enabled
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 UART2	UART2 Clock Gate Control Controls the clock gate to the UART2 module. 0 Clock disabled 1 Clock enabled
11 UART1	UART1 Clock Gate Control Controls the clock gate to the UART1 module. 0 Clock disabled 1 Clock enabled
10 UART0	UART0 Clock Gate Control Controls the clock gate to the UART0 module. 0 Clock disabled 1 Clock enabled
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 I2C1	I2C1 Clock Gate Control Controls the clock gate to the I^2C1 module. 0 Clock disabled 1 Clock enabled
6 I2C0	I2C0 Clock Gate Control Controls the clock gate to the I^2C0 module. 0 Clock disabled 1 Clock enabled

Table continues on the next page...

SIM_SCGC4 field descriptions (continued)

Field	Description
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
3–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

8.2.8 System Clock Gating Control Register 5 (SIM_SCGC5)

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

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

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

SIM_SCGC5 field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 PORTE	Port E Clock Gate Control Controls the clock gate to the Port E module. 0 Clock disabled 1 Clock enabled

Table continues on the next page...

SIM_SCGC5 field descriptions (continued)

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

8.2.9 System Clock Gating Control Register 6 (SIM_SCGC6)

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

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	I2S	0													DMAMUX	FTF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SIM_SCGC6 field descriptions

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

Table continues on the next page...

SIM_SCGC6 field descriptions (continued)

Field	Description
25 TPM1	TPM1 Clock Gate Control Controls the clock gate to the TPM1 module. 0 Clock disabled 1 Clock enabled
24 TPM0	TPM0 Clock Gate Control Controls the clock gate to the TPM0 module. 0 Clock disabled 1 Clock enabled
23 PIT	PIT Clock Gate Control This bit controls the clock gate to the PIT module. 0 Clock disabled 1 Clock enabled
22–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 I2S	I2S Clock Gate Control This bit controls the clock gate to the I^2S module. 0 Clock disabled 1 Clock enabled
14–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DMAMUX	DMA Mux Clock Gate Control Controls the clock gate to the DMA Mux module. 0 Clock disabled 1 Clock enabled
0 FTF	Flash Memory Clock Gate Control Controls the clock gate to the flash memory. Flash reads are still supported while the flash memory is clock gated, but entry into low power modes is blocked. 0 Clock disabled 1 Clock enabled

8.2.10 System Clock Gating Control Register 7 (SIM_SCGC7)

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

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

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

SIM_SCGC7 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 DMA	DMA Clock Gate Control Controls the clock gate to the DMA module. 0 Clock disabled 1 Clock enabled
7–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

8.2.11 System Clock Divider Register 1 (SIM_CLKDIV1)

NOTE

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

NOTE

Reset value loaded during System Reset from FTFA_FOPT[LPBOOT] (See Table 1).

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

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

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

SIM_CLKDIV1 field descriptions

Field	Description
31–28 OUTDIV1	Clock 1 Output Divider value Sets the divide value for the core/system clock, as well as the bus/flash clocks. At the end of reset, it is loaded with 0000 (divide by one), 0001 (divide by two), 0011 (divide by four), or 0111 (divide by eight) depending on the setting of the FTFA_FOPT[LPBOOT] (See Table 1). 0000 Divide-by-1. 0001 Divide-by-2. 0010 Divide-by-3. 0011 Divide-by-4. 0100 Divide-by-5. 0101 Divide-by-6. 0110 Divide-by-7. 0111 Divide-by-8. 1000 Divide-by-9. 1001 Divide-by-10. 1010 Divide-by-11. 1011 Divide-by-12. 1100 Divide-by-13. 1101 Divide-by-14. 1110 Divide-by-15. 1111 Divide-by-16.
27–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 OUTDIV4	Clock 4 Output Divider value Sets the divide value for the bus and flash clock and is in addition to the System clock divide ratio. At the end of reset, it is loaded with 0001 (divide by 2). 000 Divide-by-1. 001 Divide-by-2. 010 Divide-by-3. 011 Divide-by-4. 100 Divide-by-5. 101 Divide-by-6. 110 Divide-by-7. 111 Divide-by-8.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

8.2.12 Flash Configuration Register 1 (SIM_FCFG1)

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

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

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

* Notes:
- PFSIZE field: Device specific value.

SIM_FCFG1 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 PFSIZE	Program Flash Size Specifies the amount of program flash memory available on the device . Undefined values are reserved. 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 1111 128 KB of program flash memory, 4 KB protection region256 KB of program flash memory, 8 KB protection region

Table continues on the next page...

SIM_FCFG1 field descriptions (continued)

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

8.2.13 Flash Configuration Register 2 (SIM_FCFG2)

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

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

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

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

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

SIM_FCFG2 field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–24 MAXADDR0	Max Address lock This field concatenated with 13 trailing zeros indicates the first invalid address of program flash (block 0). For example, if MAXADDR0 = 0x10, the first invalid address of program flash (block 0) is 0x0002_0000. This would be the MAXADDR0 value for a device with 128 KB program flash in flash block 0.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
22–16 MAXADDR1	This field concatenated with leading zeros plus the value of the MAXADDR1 field indicates the first invalid address of the second program flash block (flash block 1). For example, if MAXADDR0 = MAXADDR1 = 0x10 the first invalid address of flash block 1 is 0x2_0000 + 0x2_0000. This would be the MAXADDR1 value for a device with 256 KB program flash memory across two flash blocks.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

8.2.14 Unique Identification Register Mid-High (SIM_UIDMH)

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

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

* Notes:
- UID field: Device specific value.

SIM_UIDMH field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 UID	Unique Identification Unique identification for the device.

8.2.15 Unique Identification Register Mid Low (SIM_UIDML)

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

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

* Notes:
- UID field: Device specific value.

SIM_UIDML field descriptions

Field	Description
31–0 UID	Unique Identification Unique identification for the device.

8.2.16 Unique Identification Register Low (SIM_UIDL)

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

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

* Notes:
- UID field: Device specific value.

SIM_UIDL field descriptions

Field	Description
31–0 UID	Unique Identification Unique identification for the device.

8.2.17 COP Control Register (SIM_COPC)

All of the bits in this register can be written only once after a reset.

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

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

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

SIM_COPC field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–2 COPT	COP Watchdog Timeout This write-once field selects the timeout period of the COP. COPT along with the COPCLKS field define the COP timeout period. 00 COP disabled 01 COP timeout after 2^5 LPO cycles or 2^{13} bus clock cycles 10 COP timeout after 2^8 LPO cycles or 2^{16} bus clock cycles 11 COP timeout after 2^{10} LPO cycles or 2^{18} bus clock cycles
1 COPCLKS	COP Clock Select This write-once field selects the clock source of the COP watchdog. 0 Internal 1 kHz clock is source to COP. 1 Bus clock is source to COP.
0 COPW	COP Windowed Mode Windowed mode is supported only when COP is running from the bus clock. The COP window is opened three quarters through the timeout period. 0 Normal mode 1 Windowed mode

8.2.18 Service COP (SIM_SRVCOP)

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

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

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

SIM_SRVCOP field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7–0 SRVCOP	Service COP Register Write 0x55 and then 0xAA (in that order) to reset the COP timeout counter, writing any other value will generate a system reset.

8.3 Functional description

See Introduction section.

Chapter 9
System Mode Controller (SMC)

9.1 Introduction

NOTE

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

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 and ColdFire+ MCUs for further details on using the SMC.

9.2 Modes of operation

The ARM CPU has three primary modes of operation:

- Run
- Sleep
- Deep Sleep

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

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

ARM CPU mode	MCU mode
Sleep	Wait
Deep Sleep	Stop

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

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

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

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

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

Table 9-1. Power modes

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

Table continues on the next page...

Table 9-1. Power modes (continued)

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

9.3 Memory map and register descriptions

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

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

NOTE

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

NOTE

Before executing the WFI instruction, the last register written to must be read back. This ensures that all register writes associated with setting up the low power mode being entered have completed before the MCU enters the low power mode. Failure to do this may result in the low power mode not being entered correctly.

SMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_E000	Power Mode Protection register (SMC_PMPROT)	8	R/W	00h	9.3.1/178
4007_E001	Power Mode Control register (SMC_PMCTRL)	8	R/W	00h	9.3.2/179
4007_E002	Stop Control Register (SMC_STOPCTRL)	8	R/W	03h	9.3.3/180
4007_E003	Power Mode Status register (SMC_PMSTAT)	8	R	01h	9.3.4/182

9.3.1 Power Mode Protection register (SMC_PMPROT)

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

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

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

NOTE

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

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

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

SMC_PMPROT field descriptions

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

Table continues on the next page...

SMC_PMPROT field descriptions (continued)

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

9.3.2 Power Mode Control register (SMC_PMCTRL)

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

NOTE

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

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

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

SMC_PMCTRL field descriptions

Field	Description
7 Reserved	This field is reserved. This bit is reserved for future expansion and should always be written zero.
6–5 RUNM	Run Mode Control When written, causes entry into the selected run mode. Writes to this field are blocked if the protection level has not been enabled using the PMPROT register. **NOTE:** RUNM may be set to VLPR only when PMSTAT=RUN. After being written to VLPR, RUNM should not be written back to RUN until PMSTAT=VLPR. 00 Normal Run mode (RUN) 01 Reserved 10 Very-Low-Power Run mode (VLPR) 11 Reserved
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 STOPA	Stop Aborted When set, this read-only status bit indicates an interrupt 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 successsful. 1 The previous stop mode entry was aborted.
2–0 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 VLLSx, the VLLSM field in the STOPCTRL register is used to further select the particular VLLS submode which will be entered. **NOTE:** When set to STOP, the PSTOPO bits in the STOPCTRL register can be used to select a Partial Stop mode if desired. 000 Normal Stop (STOP) 001 Reserved 010 Very-Low-Power Stop (VLPS) 011 Low-Leakage Stop (LLS) 100 Very-Low-Leakage Stop (VLLSx) 101 Reserved 110 Reseved 111 Reserved

9.3.3 Stop Control Register (SMC_STOPCTRL)

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

NOTE

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

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

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

SMC_STOPCTRL field descriptions

Field	Description
7–6 PSTOPO	Partial Stop Option These bits control whether a Partial Stop mode is entered when STOPM=STOP. When entering a Partial Stop mode from RUN mode, the PMC, MCG and flash remain fully powered, allowing the device to wakeup almost instantaneously at the expense of higher power consumption. In PSTOP2, only system clocks are gated allowing peripherals running on bus clock to remain fully functional. In PSTOP1, both system and bus clocks are gated. 00 STOP - Normal Stop mode 01 PSTOP1 - Partial Stop with both system and bus clocks disabled 10 PSTOP2 - Partial Stop with system clock disabled and bus clock enabled 11 Reserved
5 PORPO	POR Power Option This bit controls whether the POR detect circuit is enabled in VLLS0 mode. 0 POR detect circuit is enabled in VLLS0 1 POR detect circuit is disabled in VLLS0
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	This field is reserved. This bit is reserved for future expansion and should always be written zero.
2–0 VLLSM	VLLS Mode Control This field controls which VLLS sub-mode to enter if STOPM=VLLSx. 000 VLLS0 001 VLLS1 010 Reserved 011 VLLS3 100 Reserved 101 Reserved 110 Reserved 111 Reserved

9.3.4 Power Mode Status register (SMC_PMSTAT)

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

NOTE

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

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

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

SMC_PMSTAT field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 PMSTAT	**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 000_0001 Current power mode is RUN. 000_0010 Current power mode is STOP. 000_0100 Current power mode is VLPR. 000_1000 Current power mode is VLPW. 001_0000 Current power mode is VLPS. 010_0000 Current power mode is LLS. 100_0000 Current power mode is VLLS.

9.4 Functional description

9.4.1 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.

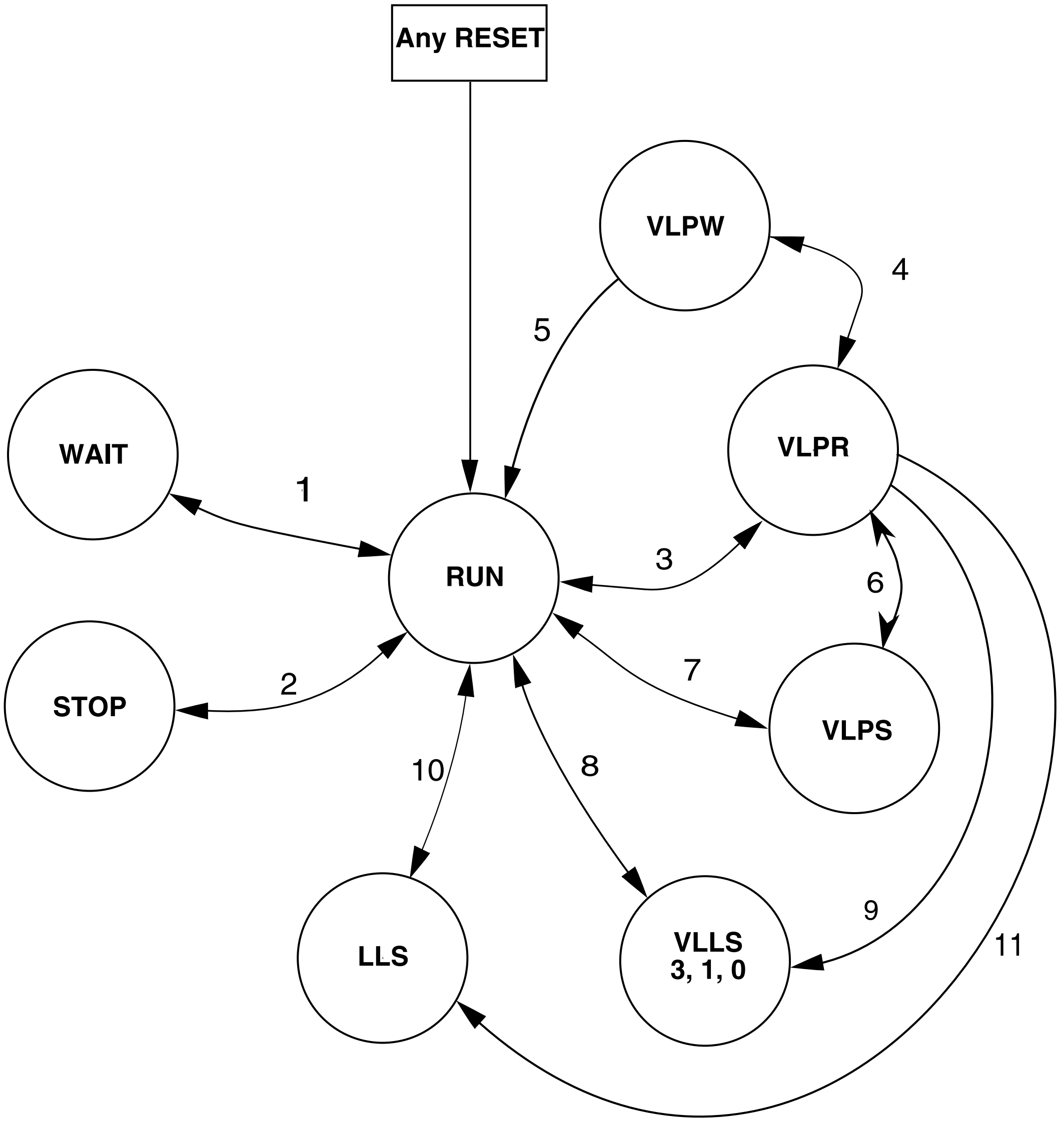


Figure 9-5. Power mode state diagram

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

Table 9-7. 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

Table continues on the next page...

Table 9-7. Power mode transition triggers (continued)

Transition #	From	To	Trigger conditions
2	RUN	STOP	PMCTRL[RUNM]=00, PMCTRL[STOPM]=000[2] Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	STOP	RUN	Interrupt or Reset
3	RUN	VLPR	The core, system, bus and flash clock frequencies and MCG clocking mode are restricted in this mode. See the Power Management chapter for the maximum allowable frequencies and MCG modes supported. Set PMPROT[AVLP]=1, PMCTRL[RUNM]=10.
	VLPR	RUN	Set PMCTRL[RUNM]=00 or Reset.
4	VLPR	VLPW	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP clear, which is controlled in System Control Register in ARM core. See note.[1]
	VLPW	VLPR	Interrupt
5	VLPW	RUN	Reset
6	VLPR	VLPS	PMCTRL[STOPM]=000[3] or 010, Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	VLPS	VLPR	Interrupt **NOTE:** If VLPS was entered directly from RUN, hardware will not allow this transition and will force exit back to RUN
7	RUN	VLPS	PMPROT[AVLP]=1, PMCTRL[STOPM]=010, Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	VLPS	RUN	Interrupt and VLPS mode was entered directly from RUN or Reset
8	RUN	VLLSx	PMPROT[AVLLS]=1, PMCTRL[STOPM]=100, STOPCTRL[VLLSM]=x (VLLSx), Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.
	VLLSx	RUN	Wakeup from enabled LLWU input source or RESET pin
9	VLPR	VLLSx	PMPROT[AVLLS]=1, PMCTRL[STOPM]=100, STOPCTRL[VLLSM]=x (VLLSx), Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.

Table continues on the next page...

Table 9-7. Power mode transition triggers (continued)

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

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

9.4.2 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.

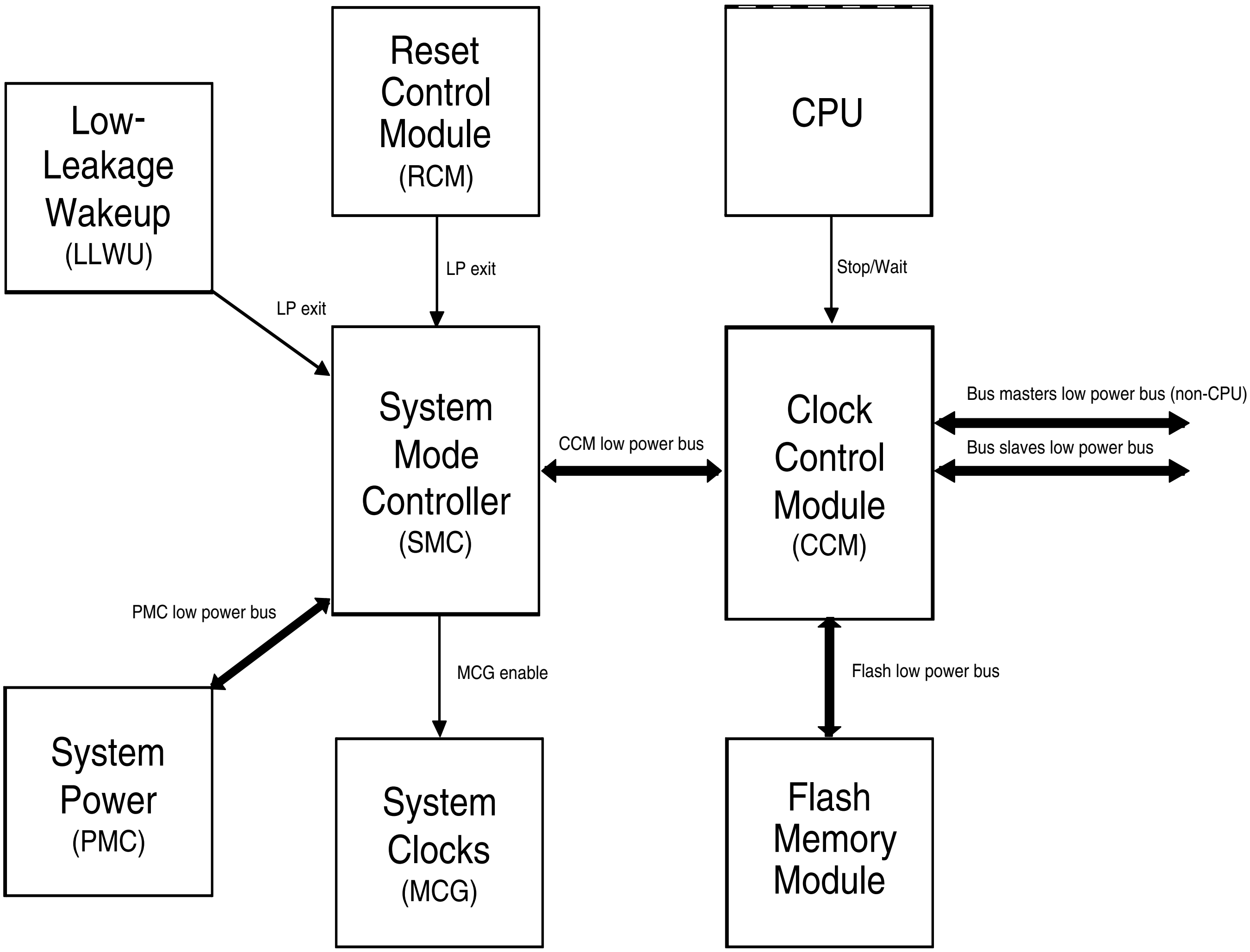


Figure 9-6. Low-power system components and connections

9.4.2.1 Stop mode entry sequence

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

1. The CPU clock is gated off immediately.

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

9.4.2.2 Stop mode exit sequence

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

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

9.4.2.3 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.

9.4.2.4 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.

9.4.2.5 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.

9.4.3 Run modes

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

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

9.4.3.1 RUN mode

This is the normal operating mode for the device.

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

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

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

9.4.3.2 Very-Low Power Run (VLPR) mode

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

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

- The MCG must be configured in a mode which is supported during VLPR. See the Power Management details for information about these MCG modes.
- All clock monitors in the MCG must be disabled.
- The maximum frequencies of the system, bus, flash, and core are restricted. See the Power Management details about which frequencies are supported.

- Mode protection must be set to allow VLP modes, that is, PMPROT[AVLP] is 1.
- PMCTRL[RUNM] is set to 10b to enter VLPR.
- Flash programming/erasing is not allowed.

NOTE

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

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

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

9.4.4 Wait modes

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

- Wait
- Very-Low Power Wait (VLPW)

9.4.4.1 WAIT mode

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

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

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

9.4.4.2 Very-Low-Power Wait (VLPW) mode

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

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

VLPR mode restrictions also apply to VLPW.

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

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

9.4.5 Stop modes

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

The stop modes range from:

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

to:

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

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

NOTE

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

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

The available stop modes are:

- Normal Stop (STOP)

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

9.4.5.1 STOP mode

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

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

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

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

9.4.5.2 Very-Low-Power Stop (VLPS) mode

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

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

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

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

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

9.4.5.3 Low-Leakage Stop (LLS) mode

Low-Leakage Stop (LLS) mode can be entered from normal RUN or VLPR modes.

The MCU enters LLS mode if:

- In Sleep-Now or Sleep-On-Exit mode, SLEEPDEEP is set in the System Control Register in the ARM core, and
- The device is configured as shown in Table 9-7.

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

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

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

NOTE

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

An asserted $\overline{\text{RESET}}$ pin will cause an exit from LLS mode, returning the device to normal RUN mode. When LLS is exiting via the $\overline{\text{RESET}}$ pin, RCM_SRS0[PIN] and RCM_SRS0[WAKEUP] are set.

9.4.5.4 Very-Low-Leakage Stop (VLLSx) modes

This device contains these very low leakage modes:

- VLLS3
- VLLS1
- VLLS0

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

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

The MCU enters the configured VLLS mode if:

- In Sleep-Now or Sleep-On-Exit mode, the SLEEPDEEP bit is set in the System Control Register in the ARM core, and
- The device is configured as shown in Table 9-7.

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

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

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

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

An asserted $\overline{\text{RESET}}$ pin will cause an exit from any VLLS mode, returning the device to normal RUN mode. When exiting VLLS via the $\overline{\text{RESET}}$ pin, RCM_SRS0[PIN] and RCM_SRS0[WAKEUP] are set.

9.4.6 Debug in low power modes

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

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

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

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

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

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

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

The VLLDBGACK bit is cleared by the debugger (or can be left set as is) or clears automatically due to the reset generated as part of the next VLLS recovery.

Chapter 10
Power Management Controller (PMC)

10.1 Introduction

NOTE

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

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

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

10.2 Features

A list of included PMC features can be found here.

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

10.3 Low-voltage detect (LVD) system

This device includes a system to guard against low-voltage conditions. This protects memory contents and controls MCU system states during supply voltage variations.

The system is comprised of a power-on reset (POR) circuit and a LVD circuit with a user-selectable trip voltage: high (V_{LVDH}) or low (V_{LVDL}). The trip voltage is selected by LVDSC1[LVDV]. The LVD is disabled upon entering VLPx, LLS, and VLLSx modes.

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

- The Low Voltage Detect Flag in the Low Voltage Status and Control 1 Register (LVDSC1[LVDF]) operates in a level sensitive manner. LVDSC1[LVDF] is set when the supply voltage falls below the selected trip point (VLVD). LVDSC1[LVDF] is cleared by writing 1 to LVDSC1[LVDACK], but only if the internal supply has returned above the trip point; otherwise, LVDSC1[LVDF] remains set.
- The Low Voltage Warning Flag (LVWF) in the Low Voltage Status and Control 2 Register (LVDSC2[LVWF]) operates in a level sensitive manner. LVDSC2[LVWF] is set when the supply voltage falls below the selected monitor trip point (VLVW). LVDSC2[LVWF] is cleared by writing one to LVDSC2[LVWACK], but only if the internal supply has returned above the trip point; otherwise, LVDSC2[LVWF] remains set.

10.3.1 LVD reset operation

By setting LVDSC1[LVDRE], the LVD generates a reset upon detection of a low-voltage condition. The low-voltage detection threshold is determined by LVDSC1[LVDV]. After an LVD reset occurs, the LVD system holds the MCU in reset until the supply voltage rises above this threshold. The LVD field in the SRS register of the RCM module (RCM_SRS0[LVD]) is set following an LVD or power-on reset.

10.3.2 LVD interrupt operation

By configuring the LVD circuit for interrupt operation (LVDSC1[LVDIE] set and LVDSC1[LVDRE] clear), LVDSC1[LVDF] is set and an LVD interrupt request occurs upon detection of a low voltage condition. LVDSC1[LVDF] is cleared by writing 1 to LVDSC1[LVDACK].

10.3.3 Low-voltage warning (LVW) interrupt operation

The LVD system contains a Low-Voltage Warning Flag (LVWF) in the Low Voltage Detect Status and Control 2 Register to indicate that the supply voltage is approaching, but is above, the LVD voltage. The LVW also has an interrupt, which is enabled by setting LVDSC2[LVWIE]. If enabled, an LVW interrupt request occurs when LVDSC2[LVWF] is set. LVDSC2[LVWF] is cleared by writing 1 to LVDSC2[LVWACK].

LVDSC2[LVWV] selects one of the four trip voltages:

- Highest: V_{LVW4}
- Two mid-levels: V_{LVW3} and V_{LVW2}
- Lowest: V_{LVW1}

10.4 I/O retention

When in LLS mode, the I/O pins are held in their input or output state.

Upon wakeup, the PMC is re-enabled, goes through a power up sequence to full regulation, and releases the logic from state retention mode. The I/O are released immediately after a wake-up or reset event. In the case of LLS exit via a RESET pin, the I/O default to their reset state.

When in VLLS modes, the I/O states are held on a wake-up event (with the exception of wake-up by reset event) until the wake-up has been acknowledged via a write to REGSC[ACKISO]. In the case of VLLS exit via a RESET pin, the I/O are released and default to their reset state. In this case, no write to REGSC[ACKISO] is needed.

10.5 Memory map and register descriptions

Details about the PMC registers can be found here.

NOTE

Different portions of PMC registers are reset only by particular reset types. Each register's description provides details. For more information about the types of reset on this chip, refer to the Reset section details.

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

PMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_D000	Low Voltage Detect Status And Control 1 register (PMC_LVDSC1)	8	R/W	10h	10.5.1/198

Table continues on the next page...

PMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_D001	Low Voltage Detect Status And Control 2 register (PMC_LVDSC2)	8	R/W	00h	10.5.2/199
4007_D002	Regulator Status And Control register (PMC_REGSC)	8	R/W	04h	10.5.3/200

10.5.1 Low Voltage Detect Status And Control 1 register (PMC_LVDSC1)

This register contains status and control bits to support the low voltage detect function. This register should be written during the reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

While the device is in the very low power or low leakage modes, the LVD system is disabled regardless of LVDSC1 settings. To protect systems that must have LVD always on, configure the Power Mode Protection (PMPROT) register of the SMC module (SMC_PMPROT) to disallow any very low power or low leakage modes from being enabled.

See the device's data sheet for the exact LVD trip voltages.

NOTE

The LVDV bits are reset solely on a POR Only event. The register's other bits are reset on Chip Reset Not VLLS. For more information about these reset types, refer to the Reset section details.

Address: 4007_D000h base + 0h offset = 4007_D000h

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

PMC_LVDSC1 field descriptions

Field	Description
7 LVDF	Low-Voltage Detect Flag This read-only status field indicates a low-voltage detect event. 0 Low-voltage event not detected 1 Low-voltage event detected

Table continues on the next page...

PMC_LVDSC1 field descriptions (continued)

Field	Description
6 LVDACK	Low-Voltage Detect Acknowledge This write-only field is used to acknowledge low voltage detection errors. Write 1 to clear LVDF. Reads always return 0.
5 LVDIE	Low-Voltage Detect Interrupt Enable Enables hardware interrupt requests for LVDF. 0 Hardware interrupt disabled (use polling) 1 Request a hardware interrupt when LVDF = 1
4 LVDRE	Low-Voltage Detect Reset Enable This write-once bit enables LVDF events to generate a hardware reset. Additional writes are ignored. 0 LVDF does not generate hardware resets 1 Force an MCU reset when LVDF = 1
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1–0 LVDV	Low-Voltage Detect Voltage Select Selects the LVD trip point voltage (V_{LVD}). 00 Low trip point selected ($V_{LVD} = V_{LVDL}$) 01 High trip point selected ($V_{LVD} = V_{LVDH}$) 10 Reserved 11 Reserved

10.5.2 Low Voltage Detect Status And Control 2 register (PMC_LVDSC2)

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

While the device is in the very low power or low leakage modes, the LVD system is disabled regardless of LVDSC2 settings.

See the device's data sheet for the exact LVD trip voltages.

NOTE

The LVW trip voltages depend on LVWV and LVDV.

NOTE

LVWV is reset solely on a POR Only event. The other fields of the register are reset on Chip Reset Not VLLS. For more information about these reset types, refer to the Reset section details.

Address: 4007_D000h base + 1h offset = 4007_D001h

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

PMC_LVDSC2 field descriptions

Field	Description
7 LVWF	Low-Voltage Warning Flag This read-only status field indicates a low-voltage warning event. LVWF is set when V_{Supply} transitions below the trip point, or after reset and V_{Supply} is already below V_{LVW}. LVWF may be 1 after power-on reset, therefore, to use LVW interrupt function, before enabling LVWIE, LVWF must be cleared by writing LVWACK first. 0 Low-voltage warning event not detected 1 Low-voltage warning event detected
6 LVWACK	Low-Voltage Warning Acknowledge This write-only field is used to acknowledge low voltage warning errors. Write 1 to clear LVWF. Reads always return 0.
5 LVWIE	Low-Voltage Warning Interrupt Enable Enables hardware interrupt requests for LVWF. 0 Hardware interrupt disabled (use polling) 1 Request a hardware interrupt when LVWF = 1
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1–0 LVWV	Low-Voltage Warning Voltage Select Selects the LVW trip point voltage (V_{LVW}). The actual voltage for the warning depends on LVDSC1[LVDV]. 00 Low trip point selected ($V_{LVW} = V_{LVW1}$) 01 Mid 1 trip point selected ($V_{LVW} = V_{LVW2}$) 10 Mid 2 trip point selected ($V_{LVW} = V_{LVW3}$) 11 High trip point selected ($V_{LVW} = V_{LVW4}$)

10.5.3 Regulator Status And Control register (PMC_REGSC)

The PMC contains an internal voltage regulator. The voltage regulator design uses a bandgap reference that is also available through a buffer as input to certain internal peripherals, such as the CMP and ADC. The internal regulator provides a status bit (REGONS) indicating the regulator is in run regulation.

NOTE

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

Address: 4007_D000h base + 2h offset = 4007_D002h

Bit	7	6	5	4	3	2	1	0
Read	0	Reserved	Reserved	BGEN	ACKISO	REGONS	Reserved	BGBE
Write					w1c			
Reset	0	0	0	0	0	1	0	0

PMC_REGSC field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved.
5 Reserved	This field is reserved.
4 BGEN	Bandgap Enable In VLPx Operation BGEN controls whether the bandgap is enabled in lower power modes of operation (VLPx, LLS, and VLLSx). When on-chip peripherals require the bandgap voltage reference in low power modes of operation, set BGEN to continue to enable the bandgap operation. **NOTE:** When the bandgap voltage reference is not needed in low power modes, clear BGEN to avoid excess power consumption. 0 Bandgap voltage reference is disabled in VLPx , LLS , and VLLSx modes. 1 Bandgap voltage reference is enabled in VLPx , LLS , and VLLSx modes.
3 ACKISO	Acknowledge Isolation Reading this field indicates whether certain peripherals and the I/O pads are in a latched state as a result of having been in a VLLS mode. Writing 1 to this field when it is set releases the I/O pads and certain peripherals to their normal run mode state. **NOTE:** After recovering from a VLLS mode, user should restore chip configuration before clearing ACKISO. In particular, pin configuration for enabled LLWU wakeup pins should be restored to avoid any LLWU flag from being falsely set when ACKISO is cleared. 0 Peripherals and I/O pads are in normal run state. 1 Certain peripherals and I/O pads are in an isolated and latched state.
2 REGONS	Regulator In Run Regulation Status This read-only field provides the current status of the internal voltage regulator. 0 Regulator is in stop regulation or in transition to/from it 1 Regulator is in run regulation
1 Reserved	This field is reserved. **NOTE:** This reserved bit must remain cleared (set to 0).

Table continues on the next page...

PMC_REGSC field descriptions (continued)

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

Chapter 11
Low-Leakage Wakeup Unit (LLWU)

11.1 Introduction

NOTE

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

The LLWU module allows the user to select up to 16 external pin sources and up to 8 internal modules as a wake-up source from low-leakage power modes.

The input sources are described in the device's chip configuration details. Each of the available wake-up sources can be individually enabled.

The LLWU module also includes two optional digital pin filters for the external wakeup pins.

See AN4503: Power Management for Kinetis and ColdFire+ MCUs for further details on using the LLWU.

11.1.1 Features

The LLWU module features include:

- Support for up to 16 external input pins and up to 8 internal modules with individual enable bits
- Input sources may be external pins or from internal peripherals capable of running in LLS or VLLS. See the chip configuration information for wakeup input sources for this device.
- External pin wake-up inputs, each of which is programmable as falling-edge, rising-edge, or any change

- Wake-up inputs that are activated if enabled after MCU enters a low-leakage power mode
- Optional digital filters provided to qualify an external pin detect. Note that when the LPO clock is disabled, the filters are disabled and bypassed.

11.1.2 Modes of operation

The LLWU module becomes functional on entry into a low-leakage power mode. After recovery from LLS, the LLWU is immediately disabled. After recovery from VLLS, the LLWU continues to detect wake-up events until the user has acknowledged the wake-up via a write to PMC_REGSC[ACKISO].

11.1.2.1 LLS mode

The wake-up events due to external wake-up inputs and internal module wake-up inputs result in an interrupt flow when exiting LLS.

NOTE

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

11.1.2.2 VLLS modes

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

11.1.2.3 Non-low leakage modes

The LLWU is not active in all non-low leakage modes where detection and control logic are in a static state. The LLWU registers are accessible in non-low leakage modes and are available for configuring and reading status when bus transactions are possible.

When the wake-up pin filters are enabled, filter operation begins immediately. If a low leakage mode is entered within five LPO clock cycles of an active edge, the edge event will be detected by the LLWU.

11.1.2.4 Debug mode

When the chip is in Debug mode and then enters LLS or a VLLSx mode, no debug logic works in the fully-functional low-leakage mode. Upon an exit from the LLS or VLLSx mode, the LLWU becomes inactive.

11.1.3 Block diagram

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

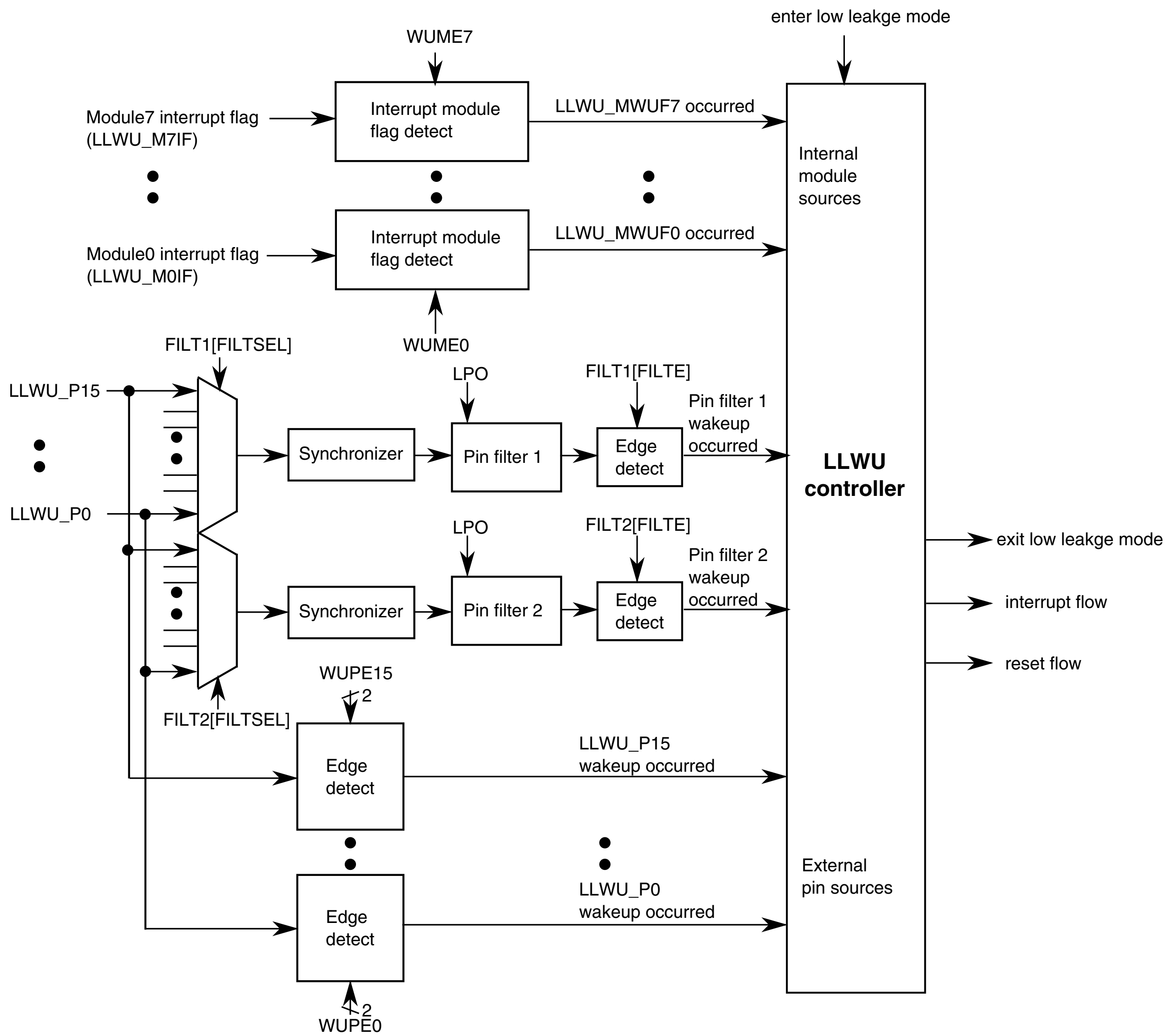


Figure 11-1. LLWU block diagram

11.2 LLWU signal descriptions

The signal properties of LLWU are shown in the table found here.

The external wakeup input pins can be enabled to detect either rising-edge, falling-edge, or on any change.

Table 11-1. LLWU signal descriptions

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

11.3 Memory map/register definition

The LLWU includes the following registers:

- Wake-up source enable registers
    - Enable external pin input sources
    - Enable internal peripheral sources
- Wake-up flag registers
    - Indication of wakeup source that caused exit from a low-leakage power mode includes external pin or internal module interrupt
- Wake-up pin filter enable registers

NOTE

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

All LLWU registers are reset by Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. Each register's displayed reset value represents this subset of reset types. LLWU registers are unaffected by reset types that do not trigger Chip Reset not VLLS. For more information about the types of reset on this chip, refer to the Introduction details.

LLWU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_C000	LLWU Pin Enable 1 register (LLWU_PE1)	8	R/W	00h	11.3.1/207
4007_C001	LLWU Pin Enable 2 register (LLWU_PE2)	8	R/W	00h	11.3.2/208
4007_C002	LLWU Pin Enable 3 register (LLWU_PE3)	8	R/W	00h	11.3.3/209
4007_C003	LLWU Pin Enable 4 register (LLWU_PE4)	8	R/W	00h	11.3.4/210
4007_C004	LLWU Module Enable register (LLWU_ME)	8	R/W	00h	11.3.5/211
4007_C005	LLWU Flag 1 register (LLWU_F1)	8	R/W	00h	11.3.6/213
4007_C006	LLWU Flag 2 register (LLWU_F2)	8	R/W	00h	11.3.7/215
4007_C007	LLWU Flag 3 register (LLWU_F3)	8	R	00h	11.3.8/216
4007_C008	LLWU Pin Filter 1 register (LLWU_FILT1)	8	R/W	00h	11.3.9/218
4007_C009	LLWU Pin Filter 2 register (LLWU_FILT2)	8	R/W	00h	11.3.10/219

11.3.1 LLWU Pin Enable 1 register (LLWU_PE1)

LLWU_PE1 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P3–LLWU_P0.

NOTE

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

Address: 4007_C000h base + 0h offset = 4007_C000h

Bit	7	6	5	4	3	2	1	0
Read / Write	WUPE3		WUPE2		WUPE1		WUPE0	
Reset	0	0	0	0	0	0	0	0

LLWU_PE1 field descriptions

Field	Description
7–6 WUPE3	Wakeup Pin Enable For LLWU_P3 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE2	Wakeup Pin Enable For LLWU_P2 Enables and configures the edge detection for the wakeup pin.

Table continues on the next page...

LLWU_PE1 field descriptions (continued)

Field	Description
	00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE1	Wakeup Pin Enable For LLWU_P1 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
1–0 WUPE0	Wakeup Pin Enable For LLWU_P0 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

11.3.2 LLWU Pin Enable 2 register (LLWU_PE2)

LLWU_PE2 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P7–LLWU_P4.

NOTE

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

Address: 4007_C000h base + 1h offset = 4007_C001h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE7		WUPE6		WUPE5		WUPE4	
Reset	0	0	0	0	0	0	0	0

LLWU_PE2 field descriptions

Field	Description
7–6 WUPE7	Wakeup Pin Enable For LLWU_P7 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection

Table continues on the next page...

LLWU_PE2 field descriptions (continued)

Field	Description
	10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE6	Wakeup Pin Enable For LLWU_P6 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE5	Wakeup Pin Enable For LLWU_P5 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
1–0 WUPE4	Wakeup Pin Enable For LLWU_P4 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

11.3.3 LLWU Pin Enable 3 register (LLWU_PE3)

LLWU_PE3 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P11–LLWU_P8.

NOTE

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

Address: 4007_C000h base + 2h offset = 4007_C002h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE11		WUPE10		WUPE9		WUPE8	
Reset	0	0	0	0	0	0	0	0

LLWU_PE3 field descriptions

Field	Description
7–6 WUPE11	Wakeup Pin Enable For LLWU_P11 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE10	Wakeup Pin Enable For LLWU_P10 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE9	Wakeup Pin Enable For LLWU_P9 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
1–0 WUPE8	Wakeup Pin Enable For LLWU_P8 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

11.3.4 LLWU Pin Enable 4 register (LLWU_PE4)

LLWU_PE4 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P15–LLWU_P12.

NOTE

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

Address: 4007_C000h base + 3h offset = 4007_C003h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE15		WUPE14		WUPE13		WUPE12	
Reset	0	0	0	0	0	0	0	0

LLWU_PE4 field descriptions

Field	Description
7–6 WUPE15	Wakeup Pin Enable For LLWU_P15 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE14	Wakeup Pin Enable For LLWU_P14 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE13	Wakeup Pin Enable For LLWU_P13 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
1–0 WUPE12	Wakeup Pin Enable For LLWU_P12 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

11.3.5 LLWU Module Enable register (LLWU_ME)

LLWU_ME contains the bits to enable the internal module flag as a wakeup input source for inputs MWUF7–MWUF0.

NOTE

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

Address: 4007_C000h base + 4h offset = 4007_C004h

Bit	7	6	5	4	3	2	1	0
Read / Write	WUME7	WUME6	WUME5	WUME4	WUME3	WUME2	WUME1	WUME0
Reset	0	0	0	0	0	0	0	0

LLWU_ME field descriptions

Field	Description
7 WUME7	Wakeup Module Enable For Module 7 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
6 WUME6	Wakeup Module Enable For Module 6 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
5 WUME5	Wakeup Module Enable For Module 5 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
4 WUME4	Wakeup Module Enable For Module 4 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
3 WUME3	Wakeup Module Enable For Module 3 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
2 WUME2	Wakeup Module Enable For Module 2 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
1 WUME1	Wakeup Module Enable for Module 1 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source

Table continues on the next page...

LLWU_ME field descriptions (continued)

Field	Description
0 WUME0	Wakeup Module Enable For Module 0 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source

11.3.6 LLWU Flag 1 register (LLWU_F1)

LLWU_F1 contains the wakeup flags indicating which wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

The external wakeup flags are read-only and clearing a flag is accomplished by a write of a 1 to the corresponding WUFx bit. The wakeup flag (WUFx), if set, will remain set if the associated WUPEx bit is cleared.

NOTE

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

Address: 4007_C000h base + 5h offset = 4007_C005h

Bit	7	6	5	4	3	2	1	0
Read	WUF7	WUF6	WUF5	WUF4	WUF3	WUF2	WUF1	WUF0
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

LLWU_F1 field descriptions

Field	Description
7 WUF7	Wakeup Flag For LLWU_P7 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF7. 0 LLWU_P7 input was not a wakeup source 1 LLWU_P7 input was a wakeup source
6 WUF6	Wakeup Flag For LLWU_P6 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF6.

Table continues on the next page...

LLWU_F1 field descriptions (continued)

Field	Description
	0 LLWU_P6 input was not a wakeup source 1 LLWU_P6 input was a wakeup source
5 WUF5	Wakeup Flag For LLWU_P5 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF5. 0 LLWU_P5 input was not a wakeup source 1 LLWU_P5 input was a wakeup source
4 WUF4	Wakeup Flag For LLWU_P4 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF4. 0 LLWU_P4 input was not a wakeup source 1 LLWU_P4 input was a wakeup source
3 WUF3	Wakeup Flag For LLWU_P3 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF3. 0 LLWU_P3 input was not a wake-up source 1 LLWU_P3 input was a wake-up source
2 WUF2	Wakeup Flag For LLWU_P2 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF2. 0 LLWU_P2 input was not a wakeup source 1 LLWU_P2 input was a wakeup source
1 WUF1	Wakeup Flag For LLWU_P1 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF1. 0 LLWU_P1 input was not a wakeup source 1 LLWU_P1 input was a wakeup source
0 WUF0	Wakeup Flag For LLWU_P0 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF0. 0 LLWU_P0 input was not a wakeup source 1 LLWU_P0 input was a wakeup source

11.3.7 LLWU Flag 2 register (LLWU_F2)

LLWU_F2 contains the wakeup flags indicating which wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

The external wakeup flags are read-only and clearing a flag is accomplished by a write of a 1 to the corresponding WUFx bit. The wakeup flag (WUFx), if set, will remain set if the associated WUPEx bit is cleared.

NOTE

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

Address: 4007_C000h base + 6h offset = 4007_C006h

Bit	7	6	5	4	3	2	1	0
Read	WUF15	WUF14	WUF13	WUF12	WUF11	WUF10	WUF9	WUF8
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

LLWU_F2 field descriptions

Field	Description
7 WUF15	Wakeup Flag For LLWU_P15 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF15. 0 LLWU_P15 input was not a wakeup source 1 LLWU_P15 input was a wakeup source
6 WUF14	Wakeup Flag For LLWU_P14 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF14. 0 LLWU_P14 input was not a wakeup source 1 LLWU_P14 input was a wakeup source
5 WUF13	Wakeup Flag For LLWU_P13 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF13. 0 LLWU_P13 input was not a wakeup source 1 LLWU_P13 input was a wakeup source

Table continues on the next page...

LLWU_F2 field descriptions (continued)

Field	Description
4 WUF12	Wakeup Flag For LLWU_P12 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF12. 0 LLWU_P12 input was not a wakeup source 1 LLWU_P12 input was a wakeup source
3 WUF11	Wakeup Flag For LLWU_P11 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF11. 0 LLWU_P11 input was not a wakeup source 1 LLWU_P11 input was a wakeup source
2 WUF10	Wakeup Flag For LLWU_P10 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF10. 0 LLWU_P10 input was not a wakeup source 1 LLWU_P10 input was a wakeup source
1 WUF9	Wakeup Flag For LLWU_P9 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF9. 0 LLWU_P9 input was not a wakeup source 1 LLWU_P9 input was a wakeup source
0 WUF8	Wakeup Flag For LLWU_P8 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF8. 0 LLWU_P8 input was not a wakeup source 1 LLWU_P8 input was a wakeup source

11.3.8 LLWU Flag 3 register (LLWU_F3)

LLWU_F3 contains the wakeup flags indicating which internal wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

For internal peripherals that are capable of running in a low-leakage power mode, such as a real time clock module or CMP module, the flag from the associated peripheral is accessible as the MWUFx bit. The flag will need to be cleared in the peripheral instead of writing a 1 to the MWUFx bit.

NOTE

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

Address: 4007_C000h base + 7h offset = 4007_C007h

Bit	7	6	5	4	3	2	1	0
Read	MWUF7	MWUF6	MWUF5	MWUF4	MWUF3	MWUF2	MWUF1	MWUF0
Write								
Reset	0	0	0	0	0	0	0	0

LLWU_F3 field descriptions

Field	Description
7 MWUF7	Wakeup flag For module 7 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 7 input was not a wakeup source 1 Module 7 input was a wakeup source
6 MWUF6	Wakeup flag For module 6 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 6 input was not a wakeup source 1 Module 6 input was a wakeup source
5 MWUF5	Wakeup flag For module 5 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 5 input was not a wakeup source 1 Module 5 input was a wakeup source
4 MWUF4	Wakeup flag For module 4 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 4 input was not a wakeup source 1 Module 4 input was a wakeup source
3 MWUF3	Wakeup flag For module 3 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 3 input was not a wakeup source 1 Module 3 input was a wakeup source
2 MWUF2	Wakeup flag For module 2

Table continues on the next page...

LLWU_F3 field descriptions (continued)

Field	Description
	Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 2 input was not a wakeup source 1 Module 2 input was a wakeup source
1 MWUF1	Wakeup flag For module 1 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 1 input was not a wakeup source 1 Module 1 input was a wakeup source
0 MWUF0	Wakeup flag For module 0 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 0 input was not a wakeup source 1 Module 0 input was a wakeup source

11.3.9 LLWU Pin Filter 1 register (LLWU_FILT1)

LLWU_FILT1 is a control and status register that is used to enable/disable the digital filter 1 features for an external pin.

NOTE

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

Address: 4007_C000h base + 8h offset = 4007_C008h

Bit	7	6	5	4	3	2	1	0
Read	FILTF	FILTE		0	FILTSEL			
Write	w1c							
Reset	0	0	0	0	0	0	0	0

LLWU_FILT1 field descriptions

Field	Description
7 FILTF	Filter Detect Flag Indicates that the filtered external wakeup pin, selected by FILTSEL, was a source of exiting a low-leakage power mode. To clear the flag write a one to FILTF.

Table continues on the next page...

LLWU_FILT1 field descriptions (continued)

Field	Description
	0 Pin Filter 1 was not a wakeup source 1 Pin Filter 1 was a wakeup source
6–5 FILTE	Digital Filter On External Pin Controls the digital filter options for the external pin detect. 00 Filter disabled 01 Filter posedge detect enabled 10 Filter negedge detect enabled 11 Filter any edge detect enabled
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 FILTSEL	Filter Pin Select Selects 1 out of the 16 wakeup pins to be muxed into the filter. 0000 Select LLWU_P0 for filter ... ... 1111 Select LLWU_P15 for filter

11.3.10 LLWU Pin Filter 2 register (LLWU_FILT2)

LLWU_FILT2 is a control and status register that is used to enable/disable the digital filter 2 features for an external pin.

NOTE

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

Address: 4007_C000h base + 9h offset = 4007_C009h

Bit	7	6	5	4	3	2	1	0
Read	FILTF	FILTE		0	FILTSEL			
Write	w1c							
Reset	0	0	0	0	0	0	0	0

LLWU_FILT2 field descriptions

Field	Description
7 FILTF	Filter Detect Flag Indicates that the filtered external wakeup pin, selected by FILTSEL, was a source of exiting a low-leakage power mode. To clear the flag write a one to FILTF.

Table continues on the next page...

LLWU_FILT2 field descriptions (continued)

Field	Description
	0 Pin Filter 2 was not a wakeup source 1 Pin Filter 2 was a wakeup source
6–5 FILTE	Digital Filter On External Pin Controls the digital filter options for the external pin detect. 00 Filter disabled 01 Filter posedge detect enabled 10 Filter negedge detect enabled 11 Filter any edge detect enabled
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 FILTSEL	Filter Pin Select Selects 1 out of the 16 wakeup pins to be muxed into the filter. 0000 Select LLWU_P0 for filter ... ... 1111 Select LLWU_P15 for filter

11.4 Functional description

This on-chip peripheral module is called a low-leakage wakeup unit (LLWU) module because it allows internal peripherals and external input pins as a source of wakeup from low-leakage modes.

It is operational only in LLS and VLLSx modes.

The LLWU module contains pin enables for each external pin and internal module. For each external pin, the user can disable or select the edge type for the wakeup. Type options are:

- Falling-edge
- Rising-edge
- Either-edge

When an external pin is enabled as a wakeup source, the pin must be configured as an input pin.

The LLWU implements optional 3-cycle glitch filters, based on the LPO clock. A detected external pin is required to remain asserted until the enabled glitch filter times out. Additional latency of up to 2 cycles is due to synchronization, which results in a total of up to 5 cycles of delay before the detect circuit alerts the system to the wakeup or reset event when the filter function is enabled. Two wakeup detect filters are available for selected external pins. Glitch filtering is not provided on the internal modules.

For internal module wakeup operation, the WUMEx bit enables the associated module as a wakeup source.

11.4.1 LLS mode

Wakeup events triggered from either an external pin input or an internal module input result in a CPU interrupt flow to begin user code execution.

11.4.2 VLLS modes

In the case of a wakeup due to external pin or internal module wakeup, recovery is always via a reset flow and RCM_SRS[WAKEUP] is set indicating the low-leakage mode was active. State retention data is lost and I/O will be restored after PMC_REGSC[ACKISO] has been written.

11.4.3 Initialization

For an enabled peripheral wakeup input, the peripheral flag must be cleared by software before entering LLS or VLLSx mode to avoid an immediate exit from the mode.

Flags associated with external input pins, filtered and unfiltered, must also be cleared by software prior to entry to LLS or VLLSx mode.

After enabling an external pin filter or changing the source pin, wait at least five LPO clock cycles before entering LLS or VLLSx mode to allow the filter to initialize.

NOTE

After recovering from a VLLS mode, user must restore chip configuration before clearing PMC_REGSC[ACKISO]. In particular, pin configuration for enabled LLWU wake-up pins must be restored to avoid any LLWU flag from being falsely set when PMC_REGSC[ACKISO] is cleared.

The signal selected as a wake-up source pin must be a digital pin, as selected in the pin mux control.

Chapter 12
Reset Control Module (RCM)

12.1 Introduction

Information found here describes the registers of the Reset Control Module (RCM). The RCM implements many of the reset functions for the chip. See the chip's reset chapter for more information.

See AN4503: Power Management for Kinetis and ColdFire+ MCUs for further details on using the RCM.

12.2 Reset memory map and register descriptions

The RCM Memory Map/Register Definition can be found here.

The Reset Control Module (RCM) registers provide reset status information and reset filter control.

NOTE

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

RCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_F000	System Reset Status Register 0 (RCM_SRS0)	8	R	82h	12.2.1/224
4007_F001	System Reset Status Register 1 (RCM_SRS1)	8	R	00h	12.2.2/225
4007_F004	Reset Pin Filter Control register (RCM_RPFC)	8	R/W	00h	12.2.3/226
4007_F005	Reset Pin Filter Width register (RCM_RPFW)	8	R/W	00h	12.2.4/227

12.2.1 System Reset Status Register 0 (RCM_SRS0)

This register includes read-only status flags to indicate the source of the most recent reset. The reset state of these bits depends on what caused the MCU to reset.

NOTE

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

- POR (including LVD) — 0x82
- LVD (without POR) — 0x02
- VLLS mode wakeup due to $\overline{\text{RESET}}$ pin assertion — 0x41
- VLLS mode wakeup due to other wakeup sources — 0x01
- Other reset — a bit is set if its corresponding reset source caused the reset

Address: 4007_F000h base + 0h offset = 4007_F000h

Bit	7	6	5	4	3	2	1	0
Read	POR	PIN	WDOG	0	LOL	LOC	LVD	WAKEUP
Write								
Reset	1	0	0	0	0	0	1	0

RCM_SRS0 field descriptions

Field	Description
7 POR	Power-On Reset Indicates a reset has been caused by the power-on detection logic. Because the internal supply voltage was ramping up at the time, the low-voltage reset (LVD) status bit is also set to indicate that the reset occurred while the internal supply was below the LVD threshold. 0 Reset not caused by POR 1 Reset caused by POR
6 PIN	External Reset Pin Indicates a reset has been caused by an active-low level on the external $\overline{\text{RESET}}$ pin. 0 Reset not caused by external reset pin 1 Reset caused by external reset pin
5 WDOG	Watchdog Indicates a reset has been caused by the watchdog timer timing out. This reset source can be blocked by disabling the watchdog. 0 Reset not caused by watchdog timeout 1 Reset caused by watchdog timeout
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

RCM_SRS0 field descriptions (continued)

Field	Description
3 LOL	Loss-of-Lock Reset Indicates a reset has been caused by a loss of lock in the MCG PLL. See the MCG description for information on the loss-of-clock 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 LOC	Loss-of-Clock Reset Indicates a reset has been caused by a loss of external clock. The MCG clock monitor must be enabled for a loss of clock to be detected. Refer to the detailed MCG description for information on enabling the clock monitor. 0 Reset not caused by a loss of external clock. 1 Reset caused by a loss of external clock.
1 LVD	Low-Voltage Detect Reset If PMC_LVDSC1[LVDRE] is set and the supply drops below the LVD trip voltage, an LVD reset occurs. This field is also set by POR. 0 Reset not caused by LVD trip or POR 1 Reset caused by LVD trip or POR
0 WAKEUP	Low Leakage Wakeup Reset Indicates a reset has been caused by an enabled LLWU module wakeup source while the chip was in a low leakage mode. In LLS mode, the $\overline{\text{RESET}}$ pin is the only wakeup source that can cause this reset. Any enabled wakeup source in a VLLSx mode causes a reset. This bit is cleared by any reset except WAKEUP. 0 Reset not caused by LLWU module wakeup source 1 Reset caused by LLWU module wakeup source

12.2.2 System Reset Status Register 1 (RCM_SRS1)

This register includes read-only status flags to indicate the source of the most recent reset. The reset state of these bits depends on what caused the MCU to reset.

NOTE

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

- POR (including LVD) — 0x00
- LVD (without POR) — 0x00
- VLLS mode wakeup — 0x00
- Other reset — a bit is set if its corresponding reset source caused the reset

Address: 4007_F000h base + 1h offset = 4007_F001h

Bit	7	6	5	4	3	2	1	0
Read	0	0	SACKERR	0	MDM_AP	SW	LOCKUP	0
Write								
Reset	0	0	0	0	0	0	0	0

RCM_SRS1 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 SACKERR	Stop Mode Acknowledge Error Reset Indicates that after an attempt to enter Stop mode, a reset has been caused by a failure of one or more peripherals to acknowledge within approximately one second to enter stop mode. 0 Reset not caused by peripheral failure to acknowledge attempt to enter stop mode 1 Reset caused by peripheral failure to acknowledge attempt to enter stop mode
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 MDM_AP	MDM-AP System Reset Request Indicates a reset has been caused by the host debugger system setting of the System Reset Request bit in the MDM-AP Control Register. 0 Reset not caused by host debugger system setting of the System Reset Request bit 1 Reset caused by host debugger system setting of the System Reset Request bit
2 SW	Software Indicates a reset has been caused by software setting of SYSRESETREQ bit in Application Interrupt and Reset Control Register in the ARM core. 0 Reset not caused by software setting of SYSRESETREQ bit 1 Reset caused by software setting of SYSRESETREQ bit
1 LOCKUP	Core Lockup Indicates a reset has been caused by the ARM core indication of a LOCKUP event. 0 Reset not caused by core LOCKUP event 1 Reset caused by core LOCKUP event
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

12.2.3 Reset Pin Filter Control register (RCM_RPFC)

NOTE

The reset values of bits 2-0 are for Chip POR only. They are unaffected by other reset types.

NOTE

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

Address: 4007_F000h base + 4h offset = 4007_F004h

Bit	7	6	5	4	3	2	1	0
Read	0					RSTFLTSS	RSTFLTSRW	
Write								
Reset	0	0	0	0	0	0	0	0

RCM_RPFC field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 RSTFLTSS	Reset Pin Filter Select in Stop Mode Selects how the reset pin filter is enabled in Stop and VLPS modes , and also during LLS and VLLS modes. On exit from VLLS mode, this bit should be reconfigured before clearing PMC_REGSC[ACKISO]. 0 All filtering disabled 1 LPO clock filter enabled
1–0 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

12.2.4 Reset Pin Filter Width register (RCM_RPFW)

NOTE

The reset values of the bits in the RSTFLTSEL field are for Chip POR only. They are unaffected by other reset types.

Address: 4007_F000h base + 5h offset = 4007_F005h

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

RCM_RPFW field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 RSTFLTSEL	Reset Pin Filter Bus Clock Select Selects the reset pin bus clock filter width. 00000 Bus clock filter count is 1 00001 Bus clock filter count is 2 00010 Bus clock filter count is 3 00011 Bus clock filter count is 4 00100 Bus clock filter count is 5 00101 Bus clock filter count is 6 00110 Bus clock filter count is 7 00111 Bus clock filter count is 8 01000 Bus clock filter count is 9 01001 Bus clock filter count is 10 01010 Bus clock filter count is 11 01011 Bus clock filter count is 12 01100 Bus clock filter count is 13 01101 Bus clock filter count is 14 01110 Bus clock filter count is 15 01111 Bus clock filter count is 16 10000 Bus clock filter count is 17 10001 Bus clock filter count is 18 10010 Bus clock filter count is 19 10011 Bus clock filter count is 20 10100 Bus clock filter count is 21 10101 Bus clock filter count is 22 10110 Bus clock filter count is 23 10111 Bus clock filter count is 24 11000 Bus clock filter count is 25 11001 Bus clock filter count is 26 11010 Bus clock filter count is 27 11011 Bus clock filter count is 28 11100 Bus clock filter count is 29 11101 Bus clock filter count is 30 11110 Bus clock filter count is 31 11111 Bus clock filter count is 32

Chapter 13
Bit Manipulation Engine (BME)

13.1 Introduction

The Bit Manipulation Engine (BME) provides hardware support for atomic read-modify-write memory operations to the peripheral address space in Cortex-M0+ based microcontrollers.

This architectural capability is also known as "decorated storage" as it defines a mechanism for providing additional semantics for load and store operations to memory-mapped peripherals beyond just the reading and writing of data values to the addressed memory locations. In the BME definition, the "decoration", that is, the additional semantic information, is encoded into the peripheral address used to reference the memory.

By combining the basic load and store instructions of the ARM Cortex-M instruction set architecture (v6M, v7M) with the concept of decorated storage provided by the BME, the resulting implementation provides a robust and efficient read-modify-write capability to this class of ultra low-end microcontrollers. The resulting architectural capability defined by this core platform function is targeted at the manipulation of n-bit fields in peripheral registers and is consistent with I/O hardware addressing in the Embedded C standard. For most BME commands, a single core read or write bus cycle is converted into an atomic read-modify-write, that is, an indivisible "read followed by a write" bus sequence.

BME decorated references are only available on system bus transactions generated by the processor core and targeted at the standard 512 KB peripheral address space based at 0x4000_0000[1]. The decoration semantic is embedded into address bits[28:19], creating a 448 MB space at addresses 0x4400_0000–0x5FFF_FFFF for AIPS; these bits are stripped out of the actual address sent to the peripheral bus controller and used by the BME to define and control its operation.

1. To be perfectly accurate, the peripheral address space occupies a 516 KB region: 512 KB based at 0x4000_0000 plus a 4 KB space based at 0x400F_F000 for GPIO accesses. This organization provides compatibility with the Kinetis K Family. Attempted accesses to the memory space located between 0x4008_0000 - 0x400F_EFFF are error terminated due to an illegal address.

13.1.1 Overview

The following figure is a generic block diagram of the processor core and platform for this class of ultra low-end microcontrollers.

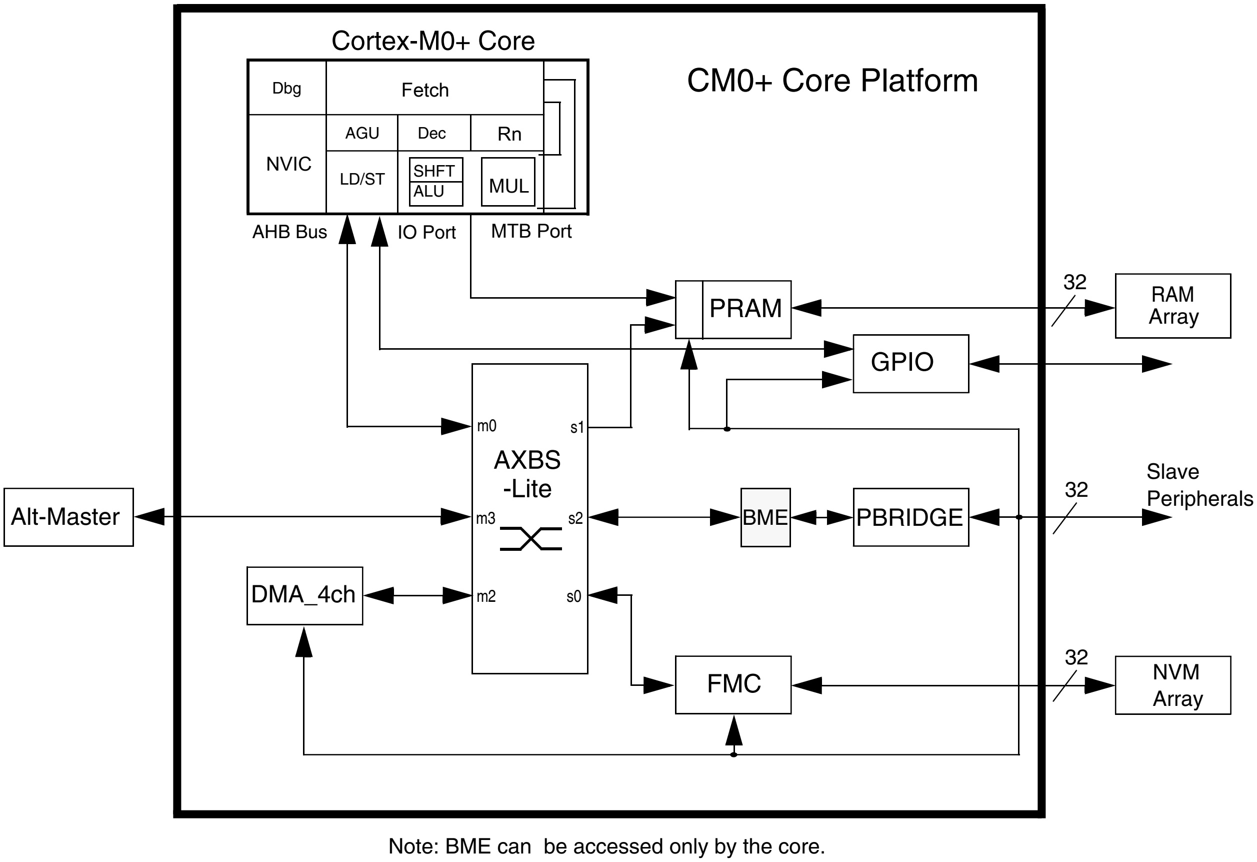


Figure 13-1. Cortex-M0+ core platform block diagram

As shown in the block diagram, the BME module interfaces to a crossbar switch AHB slave port as its primary input and sources an AHB bus output to the Peripheral Bridge (PBRIDGE) controller. The BME hardware microarchitecture is a 2-stage pipeline design matching the protocol of the AMBA-AHB system bus interfaces. The PBRIDGE module converts the AHB system bus protocol into the IPS/APB protocol used by the attached slave peripherals.

13.1.2 Features

The key features of the BME include:

- Lightweight implementation of decorated storage for selected address spaces

- Additional access semantics encoded into the reference address
- Resides between a crossbar switch slave port and a peripheral bridge bus controller
- Two-stage pipeline design matching the AHB system bus protocol
- Combinationally passes non-decorated accesses to peripheral bridge bus controller
- Conversion of decorated loads and stores from processor core into atomic read-modify-writes
- Decorated loads support unsigned bit field extracts, load-and-{set,clear} 1-bit operations
- Decorated stores support bit field inserts, logical AND, OR, and XOR operations
- Support for byte, halfword and word-sized decorated operations
- Supports minimum signal toggling on AHB output bus to reduce power dissipation

13.1.3 Modes of operation

The BME module does not support any special modes of operation. As a memory-mapped device located on a crossbar slave AHB system bus port, BME responds strictly on the basis of memory addresses for accesses to the peripheral bridge bus controller.

All functionality associated with the BME module resides in the core platform's clock domain; this includes its connections with the crossbar slave port and the PBRIDGE bus controller.

13.2 Memory map and register definition

The BME module provides a memory-mapped capability and does not include any programming model registers.

The exact set of functions supported by the BME are detailed in the Functional description.

The peripheral address space occupies a 516 KB region: 512 KB based at 0x4000_0000 plus a 4 KB space based at 0x400F_F000 for GPIO accesses; the decorated address space is mapped to the 448 MB region located at 0x4400_0000–0x5FFF_FFFF.

13.3 Functional description

Information found here details the specific functions supported by the BME.

Recall the combination of the basic load and store instructions of the Cortex-M instruction set architecture (v6M, v7M) plus the concept of decorated storage provided by the BME, the resulting implementation provides a robust and efficient read-modify-write capability to this class of ultra low-end microcontrollers. The resulting architectural capability defined by this core platform function is targeted at the manipulation of n-bit fields in peripheral registers and is consistent with I/O hardware addressing in the Embedded C standard. For most BME commands, a single core read or write bus cycle is converted into an atomic read-modify-write, that is, an indivisible "read followed by a write" bus sequence.

Consider decorated store operations first, then decorated loads.

13.3.1 BME decorated stores

The functions supported by the BME's decorated stores include three logical operators (AND, OR, XOR) plus a bit field insert.

For all these operations, BME converts a single decorated AHB store transaction into a 2-cycle atomic read-modify-write sequence, where the combined read-modify operation is performed in the first AHB data phase, and then the write is performed in the second AHB data phase.

A generic timing diagram of a decorated store showing a peripheral bit field insert operation is shown as follows:

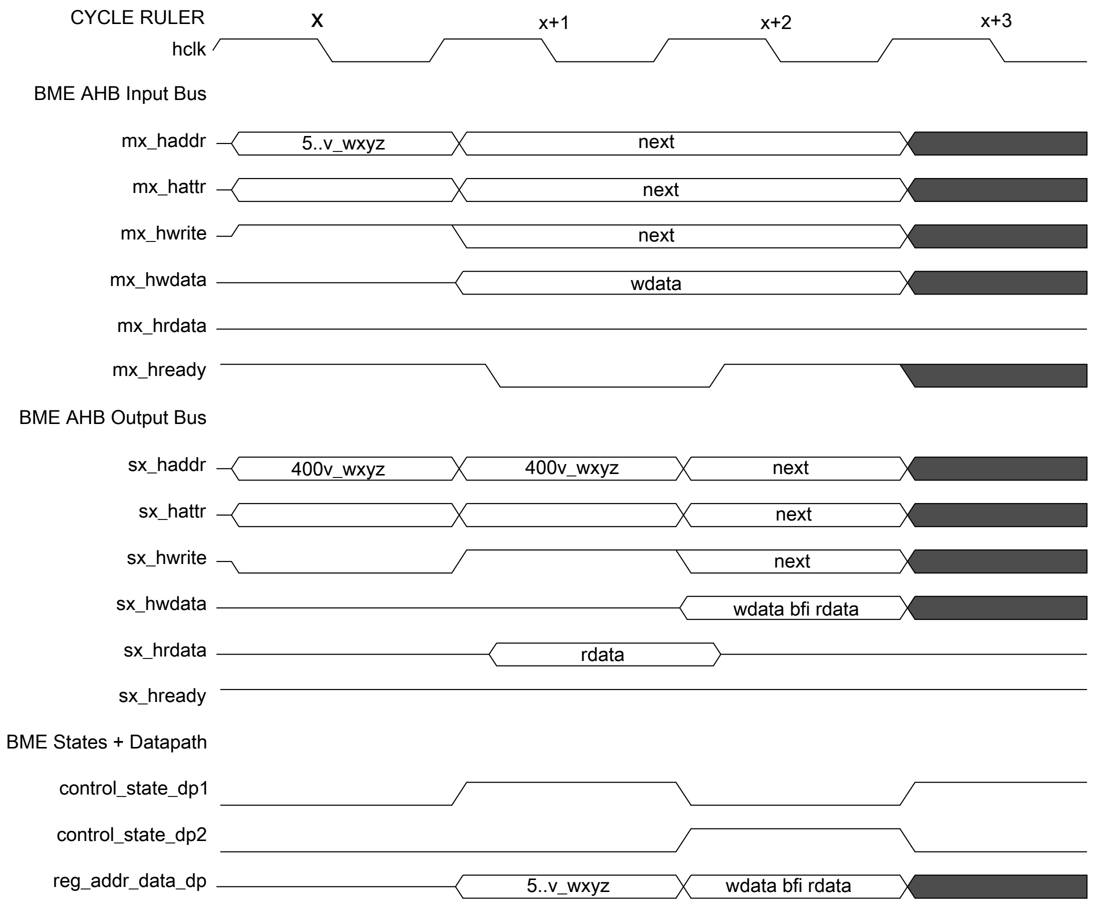


Figure 13-2. Decorated store: bit field insert timing diagram

All the decorated store operations follow the same execution template shown in Figure 13-2, a two-cycle read-modify-write operation:

1. Cycle x, 1st AHB address phase: Write from input bus is translated into a read operation on the output bus using the actual memory address (with the decoration removed) and then captured in a register.
2. Cycle x+1, 2nd AHB address phase: Write access with the registered (but actual) memory address is output
3. Cycle x+1, 1st AHB data phase: Memory read data is modified using the input bus write data and the function defined by the decoration and captured in a data register; the input bus cycle is stalled.
4. Cycle x+2, 2nd AHB data phase: Registered write data is sourced onto the output write data bus.

NOTE

Any wait states inserted by the slave device are simply passed through the BME back to the master input bus, stalling the AHB transaction cycle for cycle.

13.3.1.1 Decorated store logical AND (AND)

This command performs an atomic read-modify-write of the referenced memory location.

1. First, the location is read;
2. It is then modified by performing a logical AND operation using the write data operand sourced for the system bus cycle
3. Finally, the result of the AND operation is written back into the referenced memory location.

The data size is specified by the write operation and can be byte (8-bit), halfword (16-bit) or word (32-bit). The core performs the required write data lane replication on byte and halfword transfers.

	31	30	29	28	27	26	25	24	23	22	21	20	19–2	1	0
ioandb	0	*	*	0	0	1	-	-	-	-	-	-	mem_addr	mem_addr	mem_addr
ioandh	0	*	*	0	0	1	-	-	-	-	-	-	mem_addr	mem_addr	0
ioandw	0	*	*	0	0	1	-	-	-	-	-	-	mem_addr	0	0

Figure 13-3. Decorated store address: logical AND

See Figure 13-3 where addr[30:29] = 10 for peripheral, addr[28:26] = 001 specifies the AND operation, and mem_addr[19:0] specifies the address offset into the space based at 0x4000_0000 for peripherals. The "-" indicates an address bit "don't care".

The decorated AND write operation is defined in the following pseudo-code as:

```
ioand<sz>(accessAddress, wdata)                  // decorated store AND
tmp   =  mem[accessAddress & 0xE00FFFFF, size]  // memory read
tmp   =  tmp & wdata                             // modify
mem[accessAddress & 0xE00FFFFF, size] = tmp      // memory write
```

where the operand size <sz> is defined as b(yte, 8-bit), h(alfword, 16-bit) and w(ord, 32-bit). This notation is used throughout the document.

In the cycle definition tables, the notations AHB_ap and AHB_dp refer to the address and data phases of the BME AHB transaction. The cycle-by-cycle BME operations are detailed in the following table.

Table 13-1. Cycle definitions of decorated store: logical AND

Pipeline stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Convert master_wt to slave_rd; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form (rdata & wdata) and capture destination data in register	Perform write sending registered data to memory

13.3.1.2 Decorated store logical OR (OR)

This command performs an atomic read-modify-write of the referenced memory location.

1. First, the location is read.
2. It is then modified by performing a logical OR operation using the write data operand sourced for the system bus cycle.
3. Finally, the result of the OR operation is written back into the referenced memory location.

The data size is specified by the write operation and can be byte (8-bit), halfword (16-bit) or word (32-bit). The core performs the required write data lane replication on byte and halfword transfers.

	31	30	29	28	27	26	25	24	23	22	21	20	19–2	1	0
ioorb	0	*	*	0	1	0	-	-	-	-	-	-	mem_addr	mem_addr	mem_addr
ioorh	0	*	*	0	1	0	-	-	-	-	-	-	mem_addr	mem_addr	0
ioorw	0	*	*	0	1	0	-	-	-	-	-	-	mem_addr	0	0

Figure 13-4. Decorated address store: logical OR

See Figure 13-4,where addr[30:29] =10 for peripheral, addr[28:26] = 010 specifies the OR operation, and mem_addr[19:0] specifies the address offset into the space based at 0x4000_0000 for peripherals. The "-" indicates an address bit "don't care".

The decorated OR write operation is defined in the following pseudo-code as:

```
ioor<sz>(accessAddress, wdata)                    // decorated store OR

tmp   =  mem[accessAddress & 0xE00FFFFF, size]  // memory read
tmp   =  tmp | wdata                            // modify
mem[accessAddress & 0xE00FFFFF, size] = tmp     // memory write
```

The cycle-by-cycle BME operations are detailed in the following table.

Table 13-2. Cycle definitions of decorated store: logical OR

Pipeline stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Convert master_wt to slave_rd; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form (rdata | wdata) and capture destination data in register	Perform write sending registered data to memory

13.3.1.3 Decorated store logical XOR (XOR)

This command performs an atomic read-modify-write of the referenced memory location.

1. First, the location is read.
2. It is then modified by performing a logical XOR (exclusive-OR) operation using the write data operand sourced for the system bus cycle.
3. Finally, the result of the XOR operation is written back into the referenced memory location.

The data size is specified by the write operation and can be byte (8-bit), halfword (16-bit) or word (32-bit). The core performs the required write data lane replication on byte and halfword transfers.

	31	30	29	28	27	26	25	24	23	22	21	20	19–2	1	0
ioxorb	0	*	*	0	1	1	-	-	-	-	-	-	mem_addr		
ioxorh	0	*	*	0	1	1	-	-	-	-	-	-	mem_addr		0
ioxorw	0	*	*	0	1	1	-	-	-	-	-	-	mem_addr	0	0

Figure 13-5. Decorated address store: logical XOR

See Figure 13-5, where addr[30:29] =10 for peripheral, addr[28:26] = 011 specifies the XOR operation, and mem_addr[19:0] specifies the address offset into the peripheral space based at 0x4000_0000 for peripherals. The "-" indicates an address bit "don't care".

The decorated XOR write operation is defined in the following pseudo-code as:

```
ioxor<sz>(accessAddress, wdata)                 // decorated store XOR

tmp    =  mem[accessAddress & 0xE00FFFFF, size]  // memory read
tmp    =  tmp ^ wdata                           // modify
mem[accessAddress & 0xE00FFFFF, size] = tmp     // memory write
```

The cycle-by-cycle BME operations are detailed in the following table.

Table 13-3. Cycle definitions of decorated store: logical XOR

Pipeline Stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Convert master_wt to slave_rd; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form (rdata ^ wdata) and capture destination data in register	Perform write sending registered data to memory

13.3.1.4 Decorated store bit field insert (BFI)

This command inserts a bit field contained in the write data operand, defined by LSB position (b) and the bit field width (w+1), into the memory "container" defined by the access size associated with the store instruction using an atomic read-modify-write sequence.

The data size is specified by the write operation and can be byte (8-bit), halfword (16-bit) or word (32-bit).

NOTE

For the word sized operation, the maximum bit field width is 16 bits. The core performs the required write data lane replication on byte and halfword transfers.

The BFI operation can be used to insert a single bit into a peripheral. For this case, the w field is simply set to 0, indicating a bit field width of 1.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2	1	0
iobfib	0	*	*	1	-	-	b	b	b	-	w	w	w	mem_addr		
iobfih	0	*	*	1	-	b	b	b	b	w	w	w	w	mem_addr		0
iobfiw	0	*	*	1	b	b	b	b	b	w	w	w	w	mem_addr	0	0

Figure 13-6. Decorated address store: bit field insert

where addr[30:29] =10 for peripheral,addr[28] = 1 signals a BFI operation, addr[27:23] is "b", the LSB identifier, addr[22:19] is "w", the bit field width minus 1 identifier, and addr[18:0] specifies the address offset into the peripheral space based at 0x4000_0000 for peripherals. The "-" indicates an address bit "don't care". Note, unlike the other decorated store operations, BFI uses addr[19] as the least significant bit in the "w" specifier and not as an address bit.

The decorated BFI write operation is defined in the following pseudo-code as:

```
iobfi<sz>(accessAddress, wdata)                 // decorated bit field insert

tmp   =  mem[accessAddress & 0xE007FFFF, size]  // memory read
mask  = ((1 << (w+1)) - 1) << b                 // generate bit mask
tmp   =  tmp   & ~mask                          // modify
      |  wdata &  mask
mem[accessAddress & 0xE007FFFF, size] = tmp     // memory write
```

The write data operand (wdata) associated with the store instruction contains the bit field to be inserted. It must be properly aligned within a right-aligned container, that is, within the lower 8 bits for a byte operation, the lower 16 bits for a halfword, or the entire 32 bits for a word operation.

To illustrate, consider the following example of the insertion of the 3-bit field "xyz" into an 8-bit memory container, initially set to "abcd_efgh". For all cases, w is 2, signaling a bit field width of 3.

```
if b = 0 and the decorated store (strb) Rt register[7:0] = ----_-xyz,
   then destination is "abcd_exyz"
if b = 1 and the decorated store (strb) Rt register[7:0] = ----_xyz-,
   then destination is "abcd_xyzh"
if b = 2 and the decorated store (strb) Rt register[7:0] = ---x_yz--,
   then destination is "abcx_yzgh"
if b = 3 and the decorated store (strb) Rt register[7:0] = --xy_z---,
   then destination is "abxy_zfgh"
if b = 4 and the decorated store (strb) Rt register[7:0] = -xyz_----,
   then destination is "axyz_efgh"
if b = 5 and the decorated store (strb) Rt register[7:0] = xyz-_----,
   then destination is "xyzd_efgh"
if b = 6 and the decorated store (strb) Rt register[7:0] = yz--_----,
   then destination is "yzcd_efgh"
if b = 7 and the decorated store (strb) Rt register[7:0] = z---_----,
   then destination is "zbcd_efgh"
```

Note from the example, when the starting bit position plus the field width exceeds the container size, only part of the source bit field is inserted into the destination memory location. Stated differently, if (b + w+1) > container_width, only the low-order "container_width - b" bits are actually inserted.

The cycle-by-cycle BME operations are detailed in the following table.

Table 13-4. Cycle definitions of decorated store: bit field insert

Pipeline stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Convert master_wt to slave_rd; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form bit mask; Form bitwise ((mask) ? wdata : rdata)) and capture destination data in register	Perform write sending registered data to memory

13.3.2 BME decorated loads

The functions supported by the BME's decorated loads include two single-bit load-and-{set, clear} operators plus unsigned bit field extracts.

For the two load-and-{set, clear} operations, BME converts a single decorated AHB load transaction into a two-cycle atomic read-modify-write sequence, where the combined read-modify operations are performed in the first AHB data phase, and then the write is performed in the second AHB data phase as the original read data is returned to the processor core. For an unsigned bit field extract, the decorated load transaction is stalled for one cycle in the BME as the data field is extracted, then aligned and returned to the processor in the second AHB data phase. This is the only decorated transaction that is not an atomic read-modify-write, as it is a simple data read.

A generic timing diagram of a decorated load showing a peripheral load-and-set 1-bit operation is shown as follows.

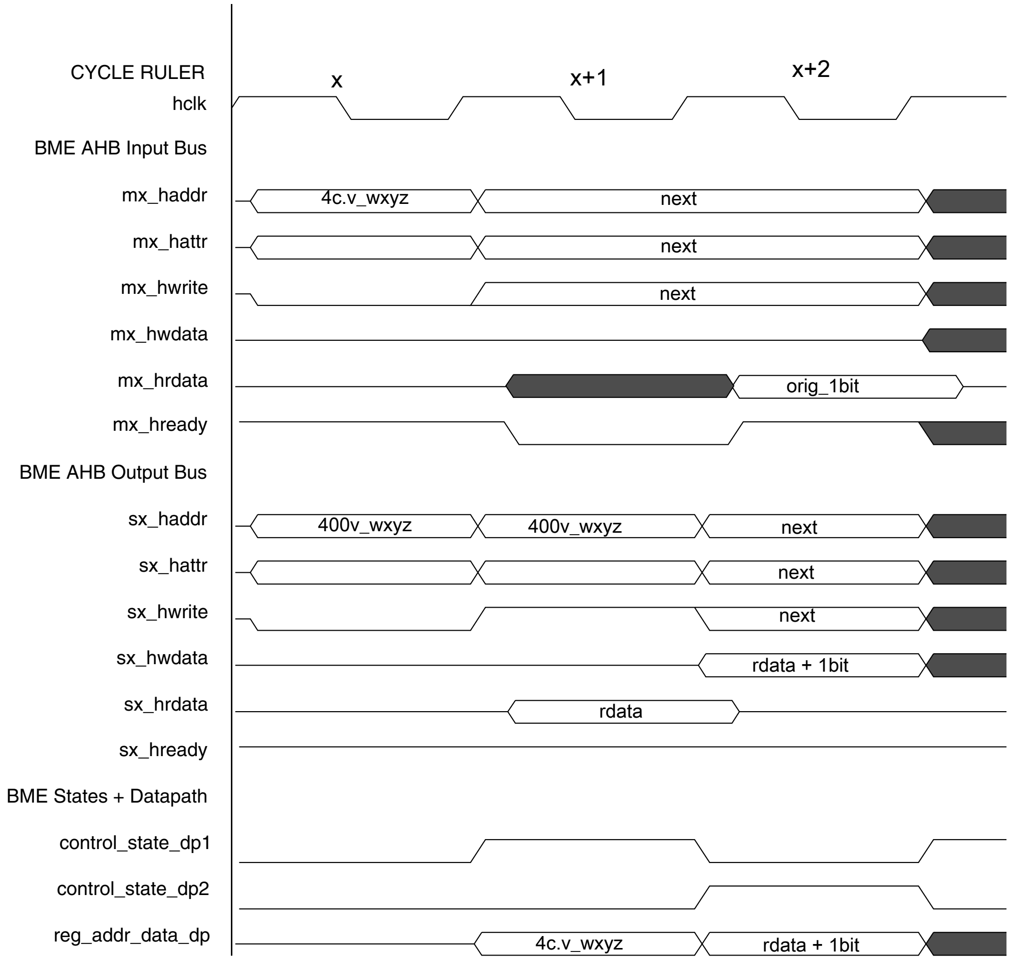


Figure 13-7. Decorated load: load-and-set 1-bit field insert timing diagram

Decorated load-and-{set, clear} 1-bit operations follow the execution template shown in the above figure: a 2-cycle read-modify-write operation:

1. Cycle x, first AHB address phase: Read from input bus is translated into a read operation on the output bus with the actual memory address (with the decoration removed) and then captured in a register
2. Cycle x+1, second AHB address phase: Write access with the registered (but actual) memory address is output
3. Cycle x+1, first AHB data phase: The "original" 1-bit memory read data is captured in a register, while the 1-bit field is set or clear based on the function defined by the decoration with the modified data captured in a register; the input bus cycle is stalled
4. Cycle x+2, second AHB data phase: The selected original 1-bit is right-justified, zero-filled and then driven onto the input read data bus, while the registered write data is sourced onto the output write data bus

NOTE

Any wait states inserted by the slave device are simply passed through the BME back to the master input bus, stalling the AHB transaction cycle for cycle.

A generic timing diagram of a decorated load showing an unsigned peripheral bit field operation is shown in the following figure.

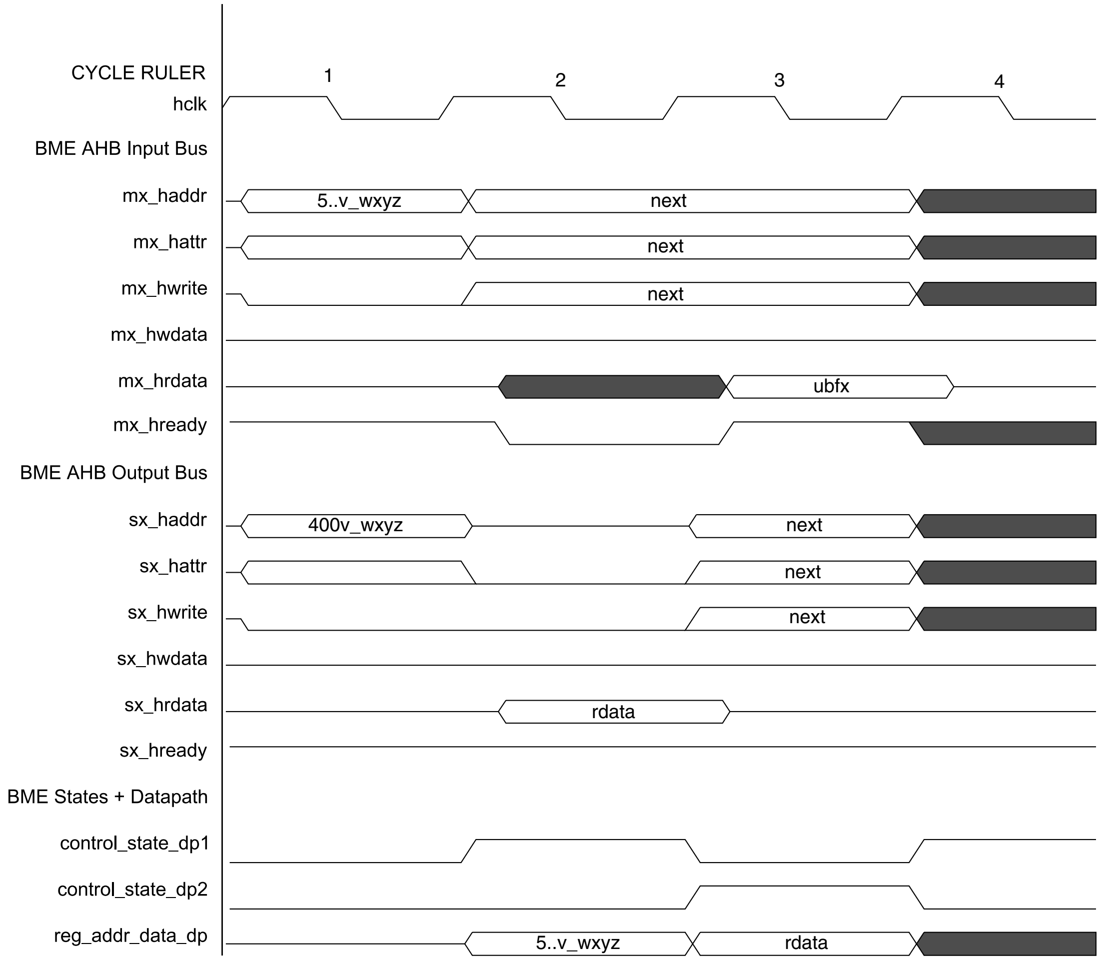


Figure 13-8. Decorated load: unsigned bit field insert timing diagram

The decorated unsigned bit field extract follows the same execution template shown in the above figure, a 2-cycle read operation:

- Cycle x, 1st AHB address phase: Read from input bus is translated into a read operation on the output bus with the actual memory address (with the decoration removed) and then captured in a register
- Cycle x+1, 2nd AHB address phase: Idle cycle

- Cycle x+1, 1st AHB data phase: A bit mask is generated based on the starting bit position and the field width; the mask is AND'ed with the memory read data to isolate the bit field; the resulting data is captured in a data register; the input bus cycle is stalled
- Cycle x+2, 2nd AHB data phase: Registered data is logically right-aligned for proper alignment and driven onto the input read data bus

NOTE

Any wait states inserted by the slave device are simply passed through the BME back to the master input bus, stalling the AHB transaction cycle for cycle.

13.3.2.1 Decorated load: load-and-clear 1 bit (LAC1)

This command loads a 1-bit field defined by the LSB position (b) into the core's general purpose destination register (Rt) and zeroes the bit in the memory space after performing an atomic read-modify-write sequence.

The extracted 1-bit data field from the memory address is right-justified and zero-filled in the operand returned to the core.

The data size is specified by the read operation and can be byte (8-bit), halfword (16-bit) or word (32-bit).

	31	30	29	28	27	26	25	24	23	22	21	20	19–2	1	0
iolac1b	0	*	*	0	1	0	-	-	b	b	b	-	mem_addr		
iolac1h	0	*	*	0	1	0	-	b	b	b	b	-	mem_addr		0
iolac1w	0	*	*	0	1	0	b	b	b	b	b	-	mem_addr	0	0

Figure 13-9. Decorated load address: load-and-clear 1 bit

See Figure 13-9 where addr[30:29] = 10 for peripheral, addr[28:26] = 010 specifies the load-and-clear 1 bit operation, addr[25:21] is "b", the bit identifier, and mem_addr[19:0] specifies the address offset into the space based at 0x4000_0000 for peripheral. The "-" indicates an address bit "don't care".

The decorated load-and-clear 1-bit read operation is defined in the following pseudo-code as:

```
rdata =  iolac1<sz>(accessAddress)              // decorated load-and-clear 1

tmp   =  mem[accessAddress & 0xE00FFFFF, size]  // memory read
mask  =  1 << b                                 // generate bit mask
rdata = (tmp &  mask) >> b                      // read data returned to core
tmp   =  tmp & ~mask                            // modify
mem[accessAddress & 0xE00FFFFF, size] = tmp     // memory write
```

The cycle-by-cycle BME operations are detailed in the following table.

Table 13-5. Cycle definitions of decorated load: load-and-clear 1 bit

Pipeline Stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form bit mask; Extract bit from rdata; Form (rdata & ~mask) and capture destination data in register	Return extracted bit to master; Perform write sending registered data to memory

13.3.2.2 Decorated Load: Load-and-Set 1 Bit (LAS1)

This command loads a 1-bit field defined by the LSB position (b) into the core's general purpose destination register (Rt) and sets the bit in the memory space after performing an atomic read-modify-write sequence.

The extracted one bit data field from the memory address is right justified and zero filled in the operand returned to the core.

The data size is specified by the read operation and can be byte (8-bit), halfword (16-bit) or word (32-bit).

	31	30	29	28	27	26	25	24	23	22	21	20	19–2	1	0
iolaslb	0	*	*	0	1	1	-	-	b	b	b	-	mem_addr		
iolaslh	0	*	*	0	1	1	-	b	b	b	b	-	mem_addr		0
iolaslw	0	*	*	0	1	1	b	b	b	b	b	-	mem_addr	0	0

Figure 13-10. Decorated load address: load-and-set 1 bit

where addr[30:29] = 10 for peripheral, addr[28:26] = 011 specifies the load-and-set 1 bit operation, addr[25:21] is "b", the bit identifier, and mem_addr[19:0] specifies the address offset into the space based at 0x4000_0000 for peripheral. The "-" indicates an address bit "don't care".

The decorated Load-and-Set 1 Bit read operation is defined in the following pseudo-code as:

```
rdata =  iolas1<sz>(accessAddress)              // decorated load-and-set 1

tmp   =  mem[accessAddress & 0xE00FFFFF, size]  // memory read
mask  =  1 << b                                 // generate bit mask
rdata = (tmp &  mask) >> b                      // read data returned to core
```

```
tmp    =  tmp |  mask                          // modify
mem[accessAddress & 0xE00FFFFF, size] = tmp    // memory write
```

The cycle-by-cycle BME operations are detailed in the following table.

Table 13-6. Cycle definitions of decorated load: load-and-set 1-bit

Pipeline Stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Capture address, attributes	Recirculate captured addr + attr to memory as slave_wt	<next>
BME AHB_dp	<previous>	Perform memory read; Form bit mask; Extract bit from rdata; Form (rdata I mask) and capture destination data in register	Return extracted bit to master; Perform write sending registered data to memory

13.3.2.3 Decorated load unsigned bit field extract (UBFX)

This command extracts a bit field defined by LSB position (b) and the bit field width (w +1) from the memory "container" defined by the access size associated with the load instruction using a two-cycle read sequence.

The extracted bit field from the memory address is right-justified and zero-filled in the operand returned to the core. Recall this is the only decorated operation that does not perform a memory write, that is, UBFX only performs a read.

The data size is specified by the write operation and can be byte (8-bit), halfword (16-bit) or word (32-bit). Note for the word sized operation, the maximum bit field width is 16 bits.

The use of a UBFX operation is recommended to extract a single bit. For this case, the w field is simply set to 0, indicating a bit field width of 1.

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>ioubfxb</td><td>0</td><td>*</td><td>*</td><td>1</td><td>-</td><td>-</td><td>b</td><td>b</td><td>b</td><td>-</td><td>w</td><td>w</td><td>w</td><td colspan="19">mem_addr</td></tr>
<tr><td>ioubfxh</td><td>0</td><td>*</td><td>*</td><td>1</td><td>-</td><td>b</td><td>b</td><td>b</td><td>b</td><td>w</td><td>w</td><td>w</td><td>w</td><td colspan="18">mem_addr</td><td>0</td></tr>
<tr><td>ioubfxw</td><td>0</td><td>*</td><td>*</td><td>1</td><td>b</td><td>b</td><td>b</td><td>b</td><td>b</td><td>w</td><td>w</td><td>w</td><td>w</td><td colspan="17">mem_addr</td><td>0</td><td>0</td></tr>
</table>

Figure 13-11. Decorated load address: unsigned bit field extract

See Figure 13-11, where addr[30:29] = 10 for peripheral, addr[28] = 1 specifies the unsigned bit field extract operation, addr[27:23] is "b", the LSB identifier, addr[22:19] is "w", the bit field width minus 1 identifier, and mem_addr[18:0] specifies the address offset into the space based at 0x4000_0000 for peripheral. The "-" indicates an address bit "don't care". Note, unlike the other decorated load operations, UBFX uses addr[19] as the least significant bit in the "w" specifier and not as an address bit.

The decorated unsigned bit field extract read operation is defined in the following pseudo-code as:

```
rdata =  ioubfx<sz>(accessAddress)              // unsigned bit field extract

tmp   =  mem[accessAddress & 0xE007FFFF, size]  // memory read
mask  = ((1 << (w+1)) - 1) << b                 // generate bit mask
rdata = (tmp &  mask) >> b                      // read data returned to core
```

Like the BFI operation, when the starting bit position plus the field width exceeds the container size, only part of the source bit field is extracted from the destination memory location. Stated differently, if (b + w+1) > container_width, only the low-order "container_width - b" bits are actually extracted. The cycle-by-cycle BME operations are detailed in the following table.

Table 13-7. Cycle definitions of decorated load: unsigned bit field extract

Pipeline Stage	Cycle		
	x	x+1	x+2
BME AHB_ap	Forward addr to memory; Decode decoration; Capture address, attributes	Idle AHB address phase	<next>
BME AHB_dp	<previous>	Perform memory read; Form bit mask; Form (rdata & mask) and capture destination data in register	Logically right shift registered data; Return justified rdata to master

13.3.3 Additional details on decorated addresses and GPIO accesses

As previously noted, the peripheral address space occupies a 516 KB region: 512 KB based at 0x4000_0000 plus a 4 KB space based at 0x400F_F000 for GPIO accesses. This memory layout provides compatibility with the Kinetis K Family and provides 129 address "slots", each 4 KB in size.

The GPIO address space is multiply-mapped by the hardware: it appears at the "standard" system address 0x400F_F000 and is physically located in the address slot corresponding to address 0x4000_F000. Decorated loads and stores create a slight complication involving accesses to the GPIO. Recall the use of address[19] varies by decorated operation; for AND, OR, XOR, LAC1 and LAS1, this bit functions as a true address bit, while for BFI and UBFX, this bit defines the least significant bit of the "w" bit field specifier.

As a result, undecorated GPIO references and decorated AND, OR, XOR, LAC1 and LAS1 operations can use the standard 0x400F_F000 base address, while decorated BFI and UBFX operations must use the alternate 0x4000_F000 base address. Another implementation can simply use 0x400F_F000 as the base address for all undecorated GPIO accesses and 0x4000_F000 as the base address for all decorated accesses. Both implementations are supported by the hardware.

Table 13-8. Decorated peripheral and GPIO address details

Peripheral address space	Description
0x4000_0000–0x4007_FFFF	Undecorated (normal) peripheral accesses
0x4008_0000–0x400F_EFFF	Illegal addresses; attempted references are aborted and error terminated
0x400F_F000–0x400F_FFFF	Undecorated (normal) GPIO accesses using standard address
0x4010_0000–0x43FF_FFFF	Illegal addresses; attempted references are aborted and error terminated
0x4400_0000–0x4FFF_FFFF	Decorated AND, OR, XOR, LAC1, LAS1 references to peripherals and GPIO based at either 0x4000_F000 or 0x400F_F000
0x5000_0000–0x5FFF_FFFF	Decorated BFI, UBFX references to peripherals and GPIO only based at 0x4000_F000

13.4 Application information

In this section, GNU assembler macros with C expression operands are presented as examples of the required instructions to perform decorated operations.

This section specifically presents a partial bme.h file defining the assembly language expressions for decorated logical stores: AND, OR, and XOR. Comparable functions for BFI and the decorated loads are more complex and available in the complete BME header file.

These macros use the same function names presented in Functional description.

```
#define IOANDW(ADDR,WDATA)                  \
    __asm("ldr    r3, =(1<<26);"     \
          "orr    r3, %[addr];"      \
          "mov    r2, %[wdata];"     \
          "str    r2, [r3];"         \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOANDH(ADDR,WDATA)                  \
    __asm("ldr    r3, =(1<<26);"     \
          "orr    r3, %[addr];"      \
          "mov    r2, %[wdata];"     \
          "strh   r2, [r3];"         \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOANDB(ADDR,WDATA)                  \
    __asm("ldr    r3, =(1<<26);"     \
          "orr    r3, %[addr];"      \
          "mov    r2, %[wdata];"     \
          "strb   r2, [r3];"         \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");
```

```
#define IOORW(ADDR,WDATA)            \
    __asm("ldr    r3, =(1<<27);"    \
          "orr    r3, %[addr];"     \
          "mov    r2, %[wdata];"    \
          "str    r2, [r3];"        \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOORH(ADDR,WDATA)            \
    __asm("ldr    r3, =(1<<27);"    \
          "orr    r3, %[addr];"     \
          "mov    r2, %[wdata];"    \
          "strh   r2, [r3];"        \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOORB(ADDR,WDATA)            \
    __asm("ldr    r3, =(1<<27);"    \
          "orr    r3, %[addr];"     \
          "mov    r2, %[wdata];"    \
          "strb   r2, [r3];"        \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOXORW(ADDR,WDATA)           \
    __asm("ldr    r3, =(3<<26);"    \
          "orr    r3, %[addr];"     \
          "mov    r2, %[wdata];"    \
          "str    r2, [r3];"        \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOXORH(ADDR,WDATA)           \
    __asm("ldr    r3, =(3<<26);"    \
          "orr    r3, %[addr];"     \
          "mov    r2, %[wdata];"    \
          "strh   r2, [r3];"        \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");

#define IOXORB(ADDR,WDATA)           \
    __asm("ldr    r3, =(3<<26);"    \
          "orr    r3, %[addr];"     \
          "mov    r2, %[wdata];"    \
          "strb   r2, [r3];"        \
          :: [addr] "r" (ADDR), [wdata] "r" (WDATA) : "r2", "r3");
```

Chapter 14
Miscellaneous Control Module (MCM)

14.1 Introduction

NOTE

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

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

14.1.1 Features

The MCM includes the following features:

- Program-visible information on the platform configuration
- Crossbar master arbitration policy selection
- Flash controller speculation buffer and cache configurations

14.2 Memory map/register descriptions

The memory map and register descriptions found here describe the registers using byte addresses. The registers can be written only when in supervisor mode.

MCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_3008	Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)	16	R	0007h	14.2.1/250

Table continues on the next page...

MCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_300A	Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)	16	R	000Dh	14.2.2/251
F000_300C	Platform Control Register (MCM_PLACR)	32	R/W	0000_0000h	14.2.3/251
F000_3040	Compute Operation Control Register (MCM_CPO)	32	R/W	0000_0000h	14.2.4/254

14.2.1 Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)

PLASC is a 16-bit read-only register identifying the presence/absence of bus slave connections to the device's crossbar switch.

Address: F000_3000h base + 8h offset = F000_3008h

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

MCM_PLASC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 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.

14.2.2 Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)

PLAMC is a 16-bit read-only register identifying the presence/absence of bus master connections to the device's crossbar switch.

Address: F000_3000h base + Ah offset = F000_300Ah

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

MCM_PLAMC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 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

14.2.3 Platform Control Register (MCM_PLACR)

The PLACR register selects the arbitration policy for the crossbar masters and configures the flash memory controller.

The speculation buffer and cache in the flash memory controller is configurable via PLACR[15:10].

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

DFCS	EFDS	Description
0	0	Speculation buffer is on for instruction and off for data.
0	1	Speculation buffer is on for instruction and on for data.
1	X	Speculation buffer is off.

The cache in flash controller is enabled and caching both instruction and data type fetches after reset. It is possible to have these states for the cache:

DFCC	DFCIC	DFCDA	Description
0	0	0	Cache is on for both instruction and data.
0	0	1	Cache is on for instruction and off for data.
0	1	0	Cache is off for instruction and on for data.
0	1	1	Cache is off for both instruction and data.
1	X	X	Cache is off.

Address: F000_3000h base + Ch offset = F000_300Ch

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DFCS	EFDS	DFCC	DFCIC	DFCDA	0	ARB	0								
W						CFCC										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCM_PLACR field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 ESFC	Enable Stalling Flash Controller Enables stalling flash controller when flash is busy.

Table continues on the next page...

MCM_PLACR field descriptions (continued)

Field	Description
	When software needs to access the flash memory while a flash memory resource is being manipulated by a flash command, software can enable a stall mechanism to avoid a read collision. The stall mechanism allows software to execute code from the same block on which flash operations are being performed. However, software must ensure the sector the flash operations are being performed on is not the same sector from which the code is executing. ESFC enables the stall mechanism. This bit must be set only just before the flash operation is executed and must be cleared when the operation completes. 0 Disable stalling flash controller when flash is busy. 1 Enable stalling flash controller when flash is busy.
15 DFCS	Disable Flash Controller Speculation Disables flash controller speculation. 0 Enable flash controller speculation. 1 Disable flash controller speculation.
14 EFDS	Enable Flash Data Speculation Enables flash data speculation. 0 Disable flash data speculation. 1 Enable flash data speculation.
13 DFCC	Disable Flash Controller Cache Disables flash controller cache. 0 Enable flash controller cache. 1 Disable flash controller cache.
12 DFCIC	Disable Flash Controller Instruction Caching Disables flash controller instruction caching. 0 Enable flash controller instruction caching. 1 Disable flash controller instruction caching.
11 DFCDA	Disable Flash Controller Data Caching Disables flash controller data caching. 0 Enable flash controller data caching 1 Disable flash controller data caching.
10 CFCC	Clear Flash Controller Cache Writing a 1 to this field clears the cache. Writing a 0 to this field is ignored. This field always reads as 0.
9 ARB	Arbitration select 0 Fixed-priority arbitration for the crossbar masters 1 Round-robin arbitration for the crossbar masters
8–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

14.2.4 Compute Operation Control Register (MCM_CPO)

This register controls the Compute Operation.

Address: F000_3000h base + 40h offset = F000_3040h

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

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

MCM_CPO field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CPOWOI	Compute Operation Wake-up on Interrupt 0 No effect. 1 When set, the CPOREQ is cleared on any interrupt or exception vector fetch.
1 CPOACK	Compute Operation Acknowledge 0 Compute operation entry has not completed or compute operation exit has completed. 1 Compute operation entry has completed or compute operation exit has not completed.
0 CPOREQ	Compute Operation Request This bit is auto-cleared by vector fetching if CPOWOI = 1.

Table continues on the next page...

MCM_CPO field descriptions (continued)

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

Chapter 15
Micro Trace Buffer (MTB)

15.1 Introduction

Microcontrollers using the Cortex-M0+ processor core include support for a CoreSight Micro Trace Buffer to provide program trace capabilities.

The proper name for this function is the CoreSight Micro Trace Buffer for the Cortex-M0+ Processor; in this document, it is simply abbreviated as the MTB.

The simple program trace function creates instruction address change-of-flow data packets in a user-defined region of the system RAM. Accordingly, the system RAM controller manages requests from two sources:

- AMBA-AHB reads and writes from the system bus
- program trace packet writes from the processor

As part of the MTB functionality, there is a DWT (Data Watchpoint and Trace) module that allows the user to define watchpoint addresses, or optionally, an address and data value, that when triggered, can be used to start or stop the program trace recording.

This document details the functionality of both the MTB_RAM and MTB_DWT capabilities.

15.1.1 Overview

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

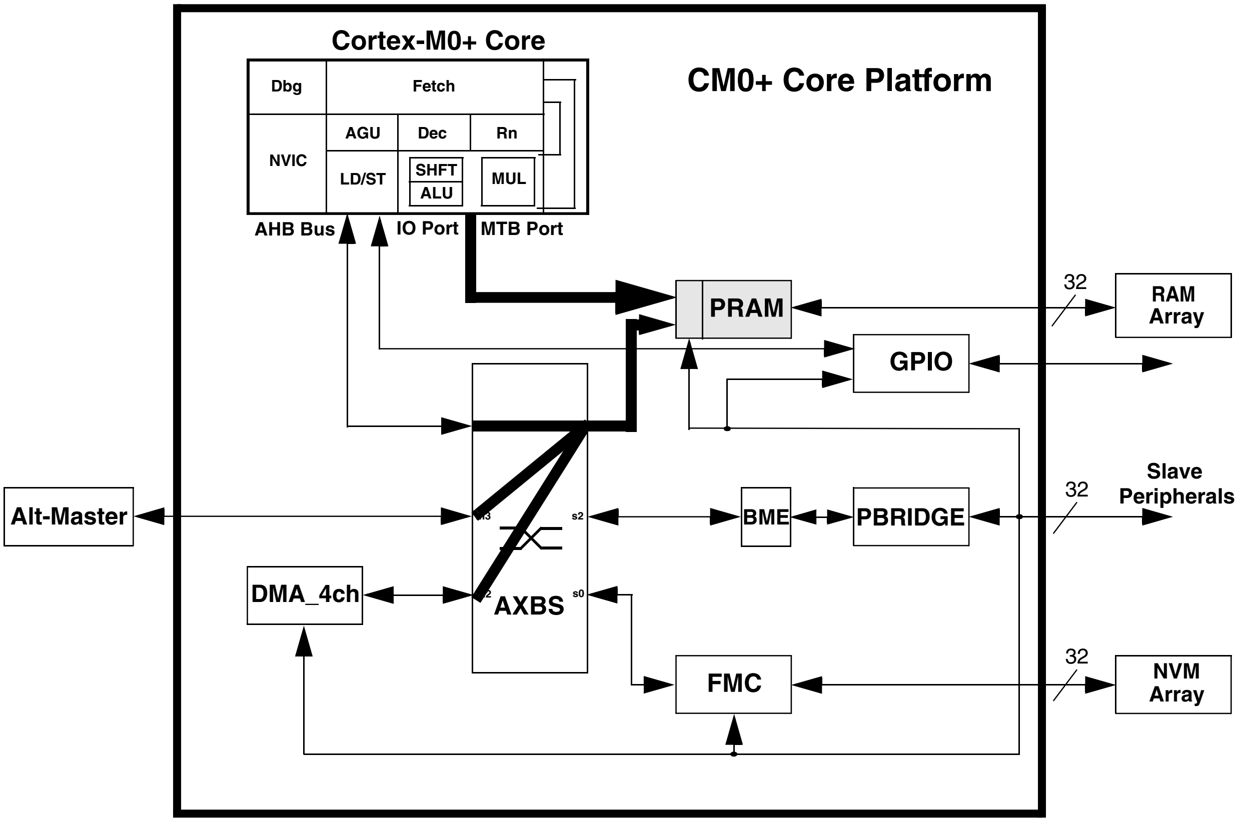


Figure 15-1. Generic Cortex-M0+ core platform block diagram

As shown in the block diagram, the platform RAM (PRAM) controller connects to two input buses:

- the crossbar slave port for system bus accesses
- a "private execution MTB port" from the core

The logical paths from the crossbar master input ports to the PRAM controller are highlighted along with the private execution trace port from the processor core. The private MTB port signals the instruction address information needed for the 64-bit program trace packets written into the system RAM. The PRAM controller output interfaces to the attached RAM array. In this document, the PRAM controller is the MTB_RAM controller.

The following information is taken from the ARM CoreSight Micro Trace Buffer documentation.

"The execution trace packet consists of a pair of 32-bit words that the MTB generates when it detects the processor PC value changes non-sequentially. A non-sequential PC change can occur during branch instructions or during exception entry.

The processor can cause a trace packet to be generated for any instruction.

The following figure shows how the execution trace information is stored in memory as a sequence of packets.

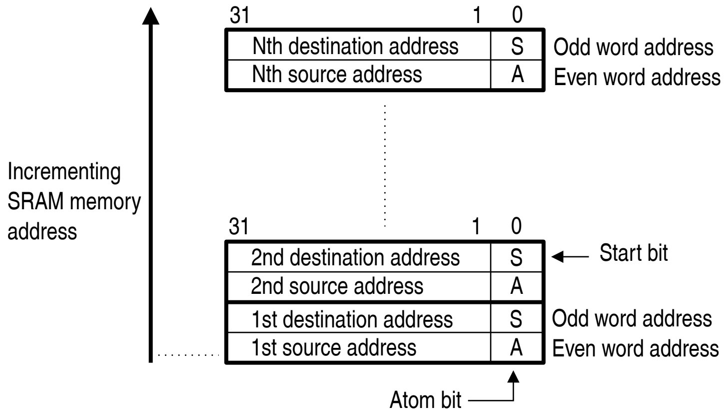


Figure 15-2. MTB execution trace storage format

The first, lower addressed, word contains the source of the branch, the address it branched from. The value stored only records bits[31:1] of the source address, because Thumb instructions are at least halfword aligned. The least significant bit of the value is the A-bit. The A-bit indicates the atomic state of the processor at the time of the branch, and can differentiate whether the branch originated from an instruction in a program, an exception, or a PC update in Debug state. When it is zero the branch originated from an instruction, when it is one the branch originated from an exception or PC update in Debug state. This word is always stored at an even word location.

The second, higher addressed word contains the destination of the branch, the address it branched to. The value stored only records bits[31:1] of the branch address. The least significant bit of the value is the S-bit. The S-bit indicates where the trace started. An S-bit value of 1 indicates where the first packet after the trace started and a value of 0 is used for other packets. Because it is possible to start and stop tracing multiple times in a trace session, the memory might contain several packets with the S-bit set to 1. This word is always stored in the next higher word in memory, an odd word address.

When the A-bit is set to 1, the source address field contains the architecturally-preferred return address for the exception. For example, if an exception was caused by an SVC instruction, then the source address field contains the address of the following instruction. This is different from the case where the A-bit is set to 0. In this case, the source address contains the address of the branch instruction.

For an exception return operation, two packets are generated:

- The first packet has the:
    - Source address field set to the address of the instruction that causes the exception return, BX or POP.

- Destination address field set to bits[31:1] of the EXC_RETURN value. See the ARM v6-M Architecture Reference Manual.
- The A-bit set to 0.

- The second packet has the:
  - Source address field set to bits[31:1] of the EXC_RETURN value.
  - Destination address field set to the address of the instruction where execution commences.
  - A-bit set to 1."

Given the recorded change-of-flow trace packets in system RAM and the memory image of the application, a debugger can read out the data and create an instruction-by-instruction program trace. In keeping with the low area and power implementation cost design targets, the MTB trace format is less efficient than other CoreSight trace modules, for example, the ETM (Embedded Trace Macrocell). Since each branch packet is 8 bytes in size, a 1 KB block of system RAM can contain 128 branches. Using the Dhrystone 2.1 benchmark's dynamic runtime as an example, this corresponds to about 875 instructions per KB of trace RAM, or with a zero wait state memory, this corresponds to approximately 1600 processor cycles per KB. This metric is obviously very sensitive to the runtime characteristics of the user code.

The MTB_DWT function (not shown in the core platform block diagram) monitors the processor address and data buses so that configurable watchpoints can be detected to trigger the appropriate response in the MTB recording.

15.1.2 Features

The key features of the MTB_RAM and MTB_DWT include:

- Memory controller for system RAM and Micro Trace Buffer for program trace packets
- Read/write capabilities for system RAM accesses, write-only for program trace packets
- Supports zero wait state response to system bus accesses when no trace data is being written
- Can buffer two AHB address phases and one data write for system RAM accesses
- Supports 64-bit program trace packets including source and destination instruction addresses
- Program trace information in RAM available to MCU's application code or external debugger
- Program trace watchpoint configuration accessible by MCU's application code or debugger
- Location and size of RAM trace buffer is configured by software

- Two DWT comparators (addresses or address + data) provide programmable start/stop recording
- CoreSight compliant debug functionality

15.1.3 Modes of operation

The MTB_RAM and MTB_DWT functions do not support any special modes of operation. The MTB_RAM controller, as a memory-mapped device located on the platform's slave AHB system bus, responds strictly on the basis of memory addresses for accesses to its attached RAM array. The MTB private execution bus provides program trace packet write information to the RAM controller. Both the MTB_RAM and MTB_DWT modules are memory-mapped, so their programming models can be accessed.

All functionality associated with the MTB_RAM and MTB_DWT modules resides in the core platform's clock domain; this includes its connections with the RAM array.

15.2 External signal description

The MTB_RAM and MTB_DWT modules do not directly support any external interfaces.

The internal interface includes a standard AHB bus with a 32-bit datapath width from the appropriate crossbar slave port plus the private execution trace bus from the processor core. The signals in the private execution trace bus are detailed in the following table taken from the ARM CoreSight Micro Trace Buffer documentation. The signal direction is defined as viewed by the MTB_RAM controller.

Table 15-1. Private execution trace port from the core to MTB_RAM

Signal	Direction	Description
LOCKUP	Input	Indicates the processor is in the Lockup state. This signal is driven LOW for cycles when the processor is executing normally and driven HIGH for every cycle the processor is waiting in the Lockup state. This signal is valid on every cycle.
IAESEQ	Input	Indicates the next instruction address in execute, IAEX, is sequential, that is non-branching.
IAEXEN	Input	IAEX register enable.
IAEX[30:0]	Input	Registered address of the instruction in the execution stage, shifted right by one bit, that is, PC >> 1.
ATOMIC	Input	Indicates the processor is performing non-instruction related activities.
EDBGRQ	Output	Request for the processor to enter the Debug state, if enabled, and halt.

In addition, there are two signals formed by the MTB_DWT module and driven to the MTB_RAM controller: TSTART (trace start) and TSTOP (trace stop). These signals can be configured using the trace watchpoints to define programmable addresses and data values to affect the program trace recording state.

15.3 Memory map and register definition

The MTB_RAM and MTB_DWT modules each support a sparsely-populated 4 KB address space for their programming models. For each address space, there are a variety of control and configurable registers near the base address, followed by a large unused address space and finally a set of CoreSight registers to support dynamic determination of the debug configuration for the device.

Accesses to the programming model follow standard ARM conventions. Taken from the ARM CoreSight Micro Trace Buffer documentation, these are:

- Do not attempt to access reserved or unused address locations. Attempting to access these locations can result in UNPREDICTABLE behavior.
- The behavior of the MTB is UNPREDICTABLE if the registers with UNKNOWN reset values are not programmed prior to enabling trace.
- Unless otherwise stated in the accompanying text:
    - Do not modify reserved register bits
    - Ignore reserved register bits on reads
    - All register bits are reset to a logic 0 by a system or power-on reset
    - Use only word size, 32-bit, transactions to access all registers

15.3.1 MTB_RAM Memory Map

MTB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_0000	MTB Position Register (MTB_POSITION)	32	R/W	Undefined	15.31.1/263
F000_0004	MTB Master Register (MTB_MASTER)	32	R/W	See section	15.31.2/265
F000_0008	MTB Flow Register (MTB_FLOW)	32	R/W	Undefined	15.31.3/267
F000_000C	MTB Base Register (MTB_BASE)	32	R	Undefined	15.31.4/269
F000_0F00	Integration Mode Control Register (MTB_MODECTRL)	32	R	0000_0000h	15.31.5/269
F000_0FA0	Claim TAG Set Register (MTB_TAGSET)	32	R	0000_0000h	15.31.6/270
F000_0FA4	Claim TAG Clear Register (MTB_TAGCLEAR)	32	R	0000_0000h	15.31.7/270
F000_0FB0	Lock Access Register (MTB_LOCKACCESS)	32	R	0000_0000h	15.31.8/271

Table continues on the next page...

MTB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_0FB4	Lock Status Register (MTB_LOCKSTAT)	32	R	0000_0000h	15.31.9/271
F000_0FB8	Authentication Status Register (MTB_AUTHSTAT)	32	R	0000_0000h	15.31.10/ 272
F000_0FBC	Device Architecture Register (MTB_DEVICEARCH)	32	R	4770_0A31h	15.31.11/ 273
F000_0FC8	Device Configuration Register (MTB_DEVICECFG)	32	R	0000_0000h	15.31.12/ 273
F000_0FCC	Device Type Identifier Register (MTB_DEVICETYPID)	32	R	0000_0031h	15.31.13/ 274
F000_0FD0	Peripheral ID Register (MTB_PERIPHID4)	32	R	See section	15.31.14/ 274
F000_0FD4	Peripheral ID Register (MTB_PERIPHID5)	32	R	See section	15.31.14/ 274
F000_0FD8	Peripheral ID Register (MTB_PERIPHID6)	32	R	See section	15.31.14/ 274
F000_0FDC	Peripheral ID Register (MTB_PERIPHID7)	32	R	See section	15.31.14/ 274
F000_0FE0	Peripheral ID Register (MTB_PERIPHID0)	32	R	See section	15.31.14/ 274
F000_0FE4	Peripheral ID Register (MTB_PERIPHID1)	32	R	See section	15.31.14/ 274
F000_0FE8	Peripheral ID Register (MTB_PERIPHID2)	32	R	See section	15.31.14/ 274
F000_0FEC	Peripheral ID Register (MTB_PERIPHID3)	32	R	See section	15.31.14/ 274
F000_0FF0	Component ID Register (MTB_COMPID0)	32	R	See section	15.31.15/ 275
F000_0FF4	Component ID Register (MTB_COMPID1)	32	R	See section	15.31.15/ 275
F000_0FF8	Component ID Register (MTB_COMPID2)	32	R	See section	15.31.15/ 275
F000_0FFC	Component ID Register (MTB_COMPID3)	32	R	See section	15.31.15/ 275

15.31.1 MTB Position Register (MTB_POSITION)

The MTB_POSITION register contains the Trace Write Address Pointer and Wrap fields. This register can be modified by the explicit programming model writes. It is also automatically updated by the MTB hardware when trace packets are being recorded.

The base address of the system RAM in the memory map dictates special consideration for the placement of the MTB. Consider the following guidelines:

For the standard configuration where the size of the MTB is ≤ 25% of the total RAM capacity, it is recommended the MTB be based at the address defined by the MTB_BASE register. The read-only MTB_BASE register is defined by the expression (0x2000_0000 - (RAM_Size/4)). For this configuration, the MTB_POSITION register is initialized to MTB_BASE & 0x0000_7FF8.

If the size of the MTB is more than 25% but less than or equal to 50% of the total RAM capacity, it is recommended the MTB be based at address 0x2000_0000. In this configuration, the MTB_POSITION register is initialized to (0x2000_0000 & 0x0000_7FF8) = 0x0000_00000.

Following these two suggested placements provides a full-featured circular memory buffer containing program trace packets.

In the unlikely event an even larger trace buffer is required, a write-once capacity of 75% of the total RAM capacity can be based at address 0x2000_0000. The MTB_POSITION register is initialized to (0x2000_0000 & 0x0000_7FF8) = 0x0000_0000. However, this configuration cannot support operation as a circular queue and instead requires the use of the MTB_FLOW[WATERMARK] capability to automatically disable tracing or halting the processor as the number of packet writes approach the buffer capacity. See the MTB_FLOW register description for more details.

Address: F000_0000h base + 0h offset = F000_0000h

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

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

* Notes:
- x = Undefined at reset.

MTB_POSITION field descriptions

Field	Description
31–3 POINTER	Trace Packet Address Pointer[28:0] Because a packet consists of two words, the POINTER field is the address of the first word of a packet. This field contains bits[31:3] of the RAM address where the next trace packet is written. Therefore, it points to an unused location and is automatically incremented.

Table continues on the next page...

MTB_POSITION field descriptions (continued)

Field	Description

	A debug agent can calculate the system memory map address for the current location in the MTB using the following "generic" equation:
	Given mtb_size = 1 << (MTB_MASTER[MASK] + 4),
	systemAddress = MTB_BASE + (((MTB_POSITION & 0xFFFF_FFF8) + (mtb_size - (MTB_BASE & (mtb_size-1)))) & 0x0000_7FF8);
	For this device, a simpler expression also applies. See the following pseudo-code:
	if ((MTB_POSITION >> 13) == 0x3) systemAddress = (0x1FFF << 16) + (0x1 << 15) + (MTB_POSITION & 0x7FF8); else systemAddress = (0x2000 << 16) + (0x0 << 15) + (MTB_POSITION & 0x7FF8);
	NOTE: The size of the RAM is parameterized and the most significant bits of the POINTER field are RAZ/WI.
	For these devices, POSITION[31:15] == POSITION[POINTER[28:12]] are RAZ/WI. Therefore, the active bits in this field are POSITION[14:3] == POSITION[POINTER[11:0]].

2 WRAP	This field is set to 1 automatically when the POINTER value wraps as determined by the MTB_MASTER[MASK] field in the MASTER Trace Control Register. A debug agent might use the WRAP field to determine whether the trace information above and below the pointer address is valid.
1–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

15.31.2 MTB Master Register (MTB_MASTER)

The MTB_MASTER register contains the main program trace enable plus other trace controls. This register can be modified by the explicit programming model writes. MTB_MASTER[EN] and MTB_MASTER[HALTREQ] fields are also automatically updated by the MTB hardware.

Before MTB_MASTER[EN] or MTB_MASTER[TSTARTEN] are set to 1, the software must initialize the MTB_POSITION and MTB_FLOW registers.

If MTB_FLOW[WATERMARK] is used to stop tracing or to halt the processor, MTB_MASTER[MASK] must still be set to a value that prevents MTB_POSITION[POINTER] from wrapping before it reaches the MTB_FLOW[WATERMARK] value.

NOTE

The format of this mask field is different than MTBDWT_MASKn[MASK].

Address: F000_0000h base + 4h offset = F000_0004h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						HALTREQ	RAMPRIV	SFRWPRIV	TSTOPEN	TSTARTEN	MASK				
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MTB_MASTER field descriptions

Field	Description
31 EN	Main Trace Enable When this field is 1, trace data is written into the RAM memory location addressed by MTB_POSITION[POINTER]. The MTB_POSITION[POINTER] value auto increments after the trace data packet is written. EN can be automatically set to 0 using the MTB_FLOW[WATERMARK] field and the MTB_FLOW[AUTOSTOP] bit. EN is automatically set to 1 if TSTARTEN is 1 and the TSTART signal is HIGH. EN is automatically set to 0 if TSTOPEN is 1 and the TSTOP signal is HIGH. **NOTE:** If EN is set to 0 because MTB_FLOW[WATERMARK] is set, then it is not automatically set to 1 if TSTARTEN is 1 and the TSTART input is HIGH. In this case, tracing can only be restarted if MTB_FLOW[WATERMARK] or MTB_POSITION[POINTER] value is changed by software.
30–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 HALTREQ	Halt Request This field is connected to the halt request signal of the trace logic, EDBGRQ. When HALTREQ is set to 1, the EDBFGRQ is asserted if DBGEN (invasive debug enable, one of the debug authentication interface signals) is also HIGH. HALTREQ can be automatically set to 1 using MTB_FLOW[WATERMARK].
8 RAMPRIV	RAM Privilege If this field is 0, then user or privileged AHB read and write accesses to the RAM are permitted. If this field is 1, then only privileged AHB read and write accesses to the RAM are permitted and user accesses are RAZ/WI. The HPROT[1] signal determines if an access is a user or privileged mode reference.
7 SFRWPRIV	Special Function Register Write Privilege If this field is 0, then user or privileged AHB read and write accesses to the MTB_RAM Special Function Registers (programming model) are permitted. If this field is 1, then only privileged write accesses are permitted; user write accesses are ignored. The HPROT[1] signal determines if an access is user or privileged. Note MTB_RAM SFR read access are not controlled by this bit and are always permitted.

Table continues on the next page...

MTB_MASTER field descriptions (continued)

Field	Description
6 TSTOPEN	Trace Stop Input Enable If this field is 1 and the TSTOP signal is HIGH, then EN is set to 0. If a trace packet is being written to memory, the write is completed before tracing is stopped.
5 TSTARTEN	Trace Start Input Enable If this field is 1 and the TSTART signal is HIGH, then EN is set to 1. Tracing continues until a stop condition occurs.
4–0 MASK	Mask This value determines the maximum size of the trace buffer in RAM. It specifies the most-significant bit of the MTB_POSITION[POINTER] field that can be updated by automatic increment. If the trace tries to advance past this power of 2, the MTB_POSITION[WRAP] bit is set to 1, the MTB_POSITION[MASK+3:3] == MTB_POSITION[POINTER[MASK:0]] bits are set to 0, and the MTB_POSITION[14:MASK+3] == MTB_POSITION[POINTER[11:MASK+1]] bits remain unchanged. This field causes the trace packet information to be stored in a circular buffer of size 2^[MASK+4] bytes, that can be positioned in memory at multiples of this size. As detailed in the MTB_POSITION description, typical "upper limits" for the MTB size are RAM_Size/4 or RAM_Size/2. Values greater than the maximum have the same effect as the maximum.

15.31.3 MTB Flow Register (MTB_FLOW)

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

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

- Changing the MTB_POSITION[POINTER] field value to point to the beginning of the trace buffer, or
- Setting MTB_FLOW[AUTOSTOP] = 0.

A debug agent can use MTB_FLOW[AUTOSTOP] to fill the trace buffer once only without halting the processor.

A debug agent can use MTB_FLOW[AUTOHALT] to fill the trace buffer once before causing the Cortex-M0+ processor to enter the Debug state. To enter Debug state, the Cortex-M0+ processor might have to perform additional branch type operations. Therefore, the MTB_FLOW[WATERMARK] field must be set below the final entry in the trace buffer region.

Address: F000_0000h base + 8h offset = F000_0008h

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

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

* Notes:
- x = Undefined at reset.

MTB_FLOW field descriptions

Field	Description
31–3 WATERMARK	WATERMARK[28:0] This field contains an address in the same format as the MTB_POSITION[POINTER] field. When MTB_POSITION[POINTER] matches the WATERMARK field value, actions defined by the AUTOHALT and AUTOSTOP bits are performed.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 AUTOHALT	If this field is 1 and WATERMARK is equal to MTB_POSITION[POINTER], then MTB_MASTER[HALTREQ] is automatically set to 1. If the DBGEN signal is HIGH, the MTB asserts this halt request to the Cortex-M0+ processor by asserting the EDBGRQ signal.
0 AUTOSTOP	If this field is 1 and WATERMARK is equal to MTB_POSITION[POINTER], then MTB_MASTER[EN] is automatically set to 0. This stops tracing.

15.31.4 MTB Base Register (MTB_BASE)

The read-only MTB_BASE Register indicates where the RAM is located in the system memory map. This register is provided to enable auto discovery of the MTB RAM location, by a debug agent and is defined by a hardware design parameter. For this device, the base address is defined by the expression: MTB_BASE[BASEADDR] = 0x2000_0000 - (RAM_Size/4)

Address: F000_0000h base + Ch offset = F000_000Ch

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

* Notes:
- x = Undefined at reset.

MTB_BASE field descriptions

Field	Description
31–0 BASEADDR	This value is defined with a hardwired signal and the expression: 0x2000_0000 - (RAM_Size/4). For example, if the total RAM capacity is 16 KB, this field is 0x1FFF_F000.

15.31.5 Integration Mode Control Register (MTB_MODECTRL)

This register enables the device to switch from a functional mode, or default behavior, into integration mode. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + F00h offset = F000_0F00h

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

MTB_MODECTRL field descriptions

Field	Description
31–0 MODECTRL	Hardwired to 0x0000_0000

15.31.6 Claim TAG Set Register (MTB_TAGSET)

The Claim Tag Set Register returns the number of bits that can be set on a read, and enables individual bits to be set on a write. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FA0h offset = F000_0FA0h

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

MTB_TAGSET field descriptions

Field	Description
31–0 TAGSET	Hardwired to 0x0000_0000

15.31.7 Claim TAG Clear Register (MTB_TAGCLEAR)

The read/write Claim Tag Clear Register is used to read the claim status on debug resources. A read indicates the claim tag status. Writing 1 to a specific bit clears the corresponding claim tag to 0. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FA4h offset = F000_0FA4h

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

MTB_TAGCLEAR field descriptions

Field	Description
31–0 TAGCLEAR	Hardwired to 0x0000_0000

15.31.8 Lock Access Register (MTB_LOCKACCESS)

The Lock Access Register enables a write access to component registers. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FB0h offset = F000_0FB0h

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

MTB_LOCKACCESS field descriptions

Field	Description
31–0 LOCKACCESS	Hardwired to 0x0000_0000

15.31.9 Lock Status Register (MTB_LOCKSTAT)

The Lock Status Register indicates the status of the lock control mechanism. This register is used in conjunction with the Lock Access Register. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FB4h offset = F000_0FB4h

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

MTB_LOCKSTAT field descriptions

Field	Description
31–0 LOCKSTAT	Hardwired to 0x0000_0000

15.31.10 Authentication Status Register (MTB_AUTHSTAT)

The Authentication Status Register reports the required security level and current status of the security enable bit pairs. Where functionality changes on a given security level, this change must be reported in this register. It is connected to specific signals used during the auto-discovery process by an external debug agent.

MTB_AUTHSTAT[3:2] indicates if nonsecure, noninvasive debug is enabled or disabled, while MTB_AUTHSTAT[1:0] indicates the enabled/disabled state of nonsecure, invasive debug. For both 2-bit fields, 0b10 indicates the functionality is disabled and 0b11 indicates it is enabled.

Address: F000_0000h base + FB8h offset = F000_0FB8h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												1	BIT2	1	BIT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTB_AUTHSTAT field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	This read-only field is reserved and always has the value 1.
2 BIT2	Connected to NIDEN or DBGEN signal.
1 Reserved	This read-only field is reserved and always has the value 1.
0 BIT0	Connected to DBGEN.

15.31.11 Device Architecture Register (MTB_DEVICEARCH)

This register indicates the device architecture. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FBCh offset = F000_0FBCh

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

MTB_DEVICEARCH field descriptions

Field	Description
31–0 DEVICEARCH	Hardwired to 0x4770_0A31.

15.31.12 Device Configuration Register (MTB_DEVICECFG)

This register indicates the device configuration. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FC8h offset = F000_0FC8h

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

MTB_DEVICECFG field descriptions

Field	Description
31–0 DEVICECFG	Hardwired to 0x0000_0000.

15.31.13 Device Type Identifier Register (MTB_DEVICETYPID)

This register indicates the device type ID. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FCCh offset = F000_0FCCh

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

MTB_DEVICETYPID field descriptions

Field	Description
31–0 DEVICETYPID	Hardwired to 0x0000_0031.

15.31.14 Peripheral ID Register (MTB_PERIPHID*n*)

These registers indicate the peripheral IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FD0h offset + (4d × i), where i=0d to 7d

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

* Notes:
- See field descriptions for the reset values.x = Undefined at reset.

MTB_PERIPHID*n* field descriptions

Field	Description
31–0 PERIPHID	Peripheral ID4 is hardwired to 0x0000_0004; ID0 to 0x0000_0032; ID1 to 0x0000_00B9; ID2 to 0x0000_000B; and all the others to 0x0000_0000.

15.31.15 Component ID Register (MTB_COMPID*n*)

These registers indicate the component IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_0000h base + FF0h offset + (4d × i), where i=0d to 3d

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

* Notes:
- See field descriptions for the reset values.x = Undefined at reset.

MTB_COMPID*n* field descriptions

Field	Description
31–0 COMPID	Component ID Component ID0 is hardwired to 0x0000_000D; ID1 to 0x0000_0090; ID2 to 0x0000_0005; ID3 to 0x0000_00B1.

15.3.2 MTB_DWT Memory Map

The MTB_DWT programming model supports a very simplified subset of the v7M debug architecture and follows the standard ARM DWT definition.

MTBDWT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_1000	MTB DWT Control Register (MTBDWT_CTRL)	32	R	2F00_0000h	15.32.1/276
F000_1020	MTB_DWT Comparator Register (MTBDWT_COMP0)	32	R/W	0000_0000h	15.32.2/277
F000_1024	MTB_DWT Comparator Mask Register (MTBDWT_MASK0)	32	R/W	0000_0000h	15.32.3/278
F000_1028	MTB_DWT Comparator Function Register 0 (MTBDWT_FCT0)	32	R/W	0000_0000h	15.32.4/279
F000_1030	MTB_DWT Comparator Register (MTBDWT_COMP1)	32	R/W	0000_0000h	15.32.2/277
F000_1034	MTB_DWT Comparator Mask Register (MTBDWT_MASK1)	32	R/W	0000_0000h	15.32.3/278
F000_1038	MTB_DWT Comparator Function Register 1 (MTBDWT_FCT1)	32	R/W	0000_0000h	15.32.5/281
F000_1200	MTB_DWT Trace Buffer Control Register (MTBDWT_TBCTRL)	32	R/W	2000_0000h	15.32.6/282
F000_1FC8	Device Configuration Register (MTBDWT_DEVICECFG)	32	R	0000_0000h	15.32.7/284

Table continues on the next page...

MTBDWT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_1FCC	Device Type Identifier Register (MTBDWT_DEVICETYPID)	32	R	0000_0004h	15.32.8/284
F000_1FD0	Peripheral ID Register (MTBDWT_PERIPHID4)	32	R	See section	15.32.9/285
F000_1FD4	Peripheral ID Register (MTBDWT_PERIPHID5)	32	R	See section	15.32.9/285
F000_1FD8	Peripheral ID Register (MTBDWT_PERIPHID6)	32	R	See section	15.32.9/285
F000_1FDC	Peripheral ID Register (MTBDWT_PERIPHID7)	32	R	See section	15.32.9/285
F000_1FE0	Peripheral ID Register (MTBDWT_PERIPHID0)	32	R	See section	15.32.9/285
F000_1FE4	Peripheral ID Register (MTBDWT_PERIPHID1)	32	R	See section	15.32.9/285
F000_1FE8	Peripheral ID Register (MTBDWT_PERIPHID2)	32	R	See section	15.32.9/285
F000_1FEC	Peripheral ID Register (MTBDWT_PERIPHID3)	32	R	See section	15.32.9/285
F000_1FF0	Component ID Register (MTBDWT_COMPID0)	32	R	See section	15.32.10/ 285
F000_1FF4	Component ID Register (MTBDWT_COMPID1)	32	R	See section	15.32.10/ 285
F000_1FF8	Component ID Register (MTBDWT_COMPID2)	32	R	See section	15.32.10/ 285
F000_1FFC	Component ID Register (MTBDWT_COMPID3)	32	R	See section	15.32.10/ 285

15.32.1 MTB DWT Control Register (MTBDWT_CTRL)

The MTBDWT_CTRL register provides read-only information on the watchpoint configuration for the MTB_DWT.

Address: F000_1000h base + 0h offset = F000_1000h

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

MTBDWT_CTRL field descriptions

Field	Description
31–28 NUMCMP	Number of comparators The MTB_DWT implements two comparators.
27–0 DWTCFGCTRL	DWT configuration controls This field is hardwired to 0xF00_0000, disabling all the remaining DWT functionality. The specific fields and their state are: MTBDWT_CTRL[27] = NOTRCPKT = 1, trace sample and exception trace is not supported

Table continues on the next page...

MTBDWT_CTRL field descriptions (continued)

Field	Description
	MTBDWT_CTRL[26] = NOEXTTRIG = 1, external match signals are not supported MTBDWT_CTRL[25] = NOCYCCNT = 1, cycle counter is not supported MTBDWT_CTRL[24] = NOPRFCNT = 1, profiling counters are not supported MTBDWT_CTRL[22] = CYCEBTENA = 0, no POSTCNT underflow packets generated MTBDWT_CTRL[21] = FOLDEVTENA = 0, no folded instruction counter overflow events MTBDWT_CTRL[20] = LSUEVTENA = 0, no LSU counter overflow events MTBDWT_CTRL[19] = SLEEPEVTENA = 0, no sleep counter overflow events MTBDWT_CTRL[18] = EXCEVTENA = 0, no exception overhead counter events MTBDWT_CTRL[17] = CPIEVTENA = 0, no CPI counter overflow events MTBDWT_CTRL[16] = EXCTRCENA = 0, generation of exception trace disabled MTBDWT_CTRL[12] = PCSAMPLENA = 0, no periodic PC sample packets generated MTBDWT_CTRL[11:10] = SYNCTAP = 0, no synchronization packets MTBDWT_CTRL[9] = CYCTAP = 0, cycle counter is not supported MTBDWT_CTRL[8:5] = POSTINIT = 0, cycle counter is not supported MTBDWT_CTRL[4:1] = POSTPRESET = 0, cycle counter is not supported MTBDWT_CTRL[0] = CYCCNTENA = 0, cycle counter is not supported

15.32.2 MTB_DWT Comparator Register (MTBDWT_COMP*n*)

The MTBDWT_COMPn registers provide the reference value for comparator n.

Address: F000_1000h base + 20h offset + (16d × i), where i=0d to 1d

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

MTBDWT_COMP*n* field descriptions

Field	Description
31–0 COMP	Reference value for comparison If MTBDWT_COMP0 is used for a data value comparator and the access size is byte or halfword, the data value must be replicated across all appropriate byte lanes of this register. For example, if the data is a byte-sized "x" value, then COMP[31:24] = COMP[23:16] = COMP[15:8] = COMP[7:0] = "x". Likewise, if the data is a halfword-size "y" value, then COMP[31:16] = COMP[15:0] = "y".

15.32.3 MTB_DWT Comparator Mask Register (MTBDWT_MASK*n*)

The MTBDWT_MASKn registers define the size of the ignore mask applied to the reference address for address range matching by comparator n. Note the format of this mask field is different than the MTB_MASTER[MASK].

Address: F000_1000h base + 24h offset + (16d × i), where i=0d to 1d

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

MTBDWT_MASK*n* field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 MASK	MASK The value of the ignore mask, 0-31 bits, is applied to address range matching. MASK = 0 is used to include all bits of the address in the comparison, except if MASK = 0 and the comparator is configured to watch instruction fetch addresses, address bit [0] is ignored by the hardware since all fetches must be at least halfword aligned. For MASK != 0 and regardless of watch type, address bits [x-1:0] are ignored in the address comparison. Using a mask means the comparator matches on a range of addresses, defined by the unmasked most significant bits of the address, bits [31:x]. The maximum MASK value is 24, producing a 16 Mbyte mask. An attempted write of a MASK value > 24 is limited by the MTBDWT hardware to 24. If MTBDWT_COMP0 is used as a data value comparator, then MTBDWT_MASK0 should be programmed to zero.

15.32.4 MTB_DWT Comparator Function Register 0 (MTBDWT_FCT0)

The MTBDWT_FCTn registers control the operation of comparator n.

Address: F000_1000h base + 28h offset = F000_1028h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATAVADDR0				DATAVSIZE		0	DATAVMATCH	0				FUNCTION			
W	DATAVADDR0				DATAVSIZE			DATAVMATCH					FUNCTION			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTBDWT_FCT0 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 MATCHED	Comparator match If this read-only flag is asserted, it indicates the operation defined by the FUNCTION field occurred since the last read of the register. Reading the register clears this bit.

Table continues on the next page...

MTBDWT_FCT0 field descriptions (continued)

Field	Description
	0 No match. 1 Match occurred.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 DATAVADDR0	Data Value Address 0 Since the MTB_DWT implements two comparators, the DATAVADDR0 field is restricted to values {0,1}. When the DATAVMATCH bit is asserted, this field defines the comparator number to use for linked address comparison. If MTBDWT_COMP0 is used as a data watchpoint and MTBDWT_COMP1 as an address watchpoint, DATAVADDR0 must be set.
11–10 DATAVSIZE	Data Value Size For data value matching, this field defines the size of the required data comparison. 00 Byte. 01 Halfword. 10 Word. 11 Reserved. Any attempts to use this value results in UNPREDICTABLE behavior.
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 DATAVMATCH	Data Value Match When this field is 1, it enables data value comparison. For this implementation, MTBDWT_COMP0 supports address or data value comparisons; MTBDWT_COMP1 only supports address comparisons. 0 Perform address comparison. 1 Perform data value comparison.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 FUNCTION	Function Selects the action taken on a comparator match. If MTBDWT_COMP0 is used for a data value and MTBDWT_COMP1 for an address value, then MTBDWT_FCT1[FUNCTION] must be set to zero. For this configuration, MTBDWT_MASK1 can be set to a non-zero value, so the combined comparators match on a range of addresses. 0000 Disabled. 0100 Instruction fetch. 0101 Data operand read. 0110 Data operand write. 0111 Data operand (read + write). others Reserved. Any attempts to use this value results in UNPREDICTABLE behavior.

15.32.5 MTB_DWT Comparator Function Register 1 (MTBDWT_FCT1)

The MTBDWT_FCTn registers control the operation of comparator n. Since the MTB_DWT only supports data value comparisons on comparator 0, there are several fields in the MTBDWT_FCT1 register that are RAZ/WI (bits 12, 11:10, 8).

Address: F000_1000h base + 38h offset = F000_1038h

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

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

MTBDWT_FCT1 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 MATCHED	Comparator match If this read-only flag is asserted, it indicates the operation defined by the FUNCTION field occurred since the last read of the register. Reading the register clears this bit.

Table continues on the next page...

MTBDWT_FCT1 field descriptions (continued)

Field	Description
	0 No match. 1 Match occurred.
23–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 FUNCTION	Function Selects the action taken on a comparator match. If MTBDWT_COMP0 is used for a data value and MTBDWT_COMP1 for an address value, then MTBDWT_FCT1[FUNCTION] must be set to zero. For this configuration, MTBDWT_MASK1 can be set to a non-zero value, so the combined comparators match on a range of addresses. 0000 Disabled. 0100 Instruction fetch. 0101 Data operand read. 0110 Data operand write. 0111 Data operand (read + write). others Reserved. Any attempts to use this value results in UNPREDICTABLE behavior.

15.32.6 MTB_DWT Trace Buffer Control Register (MTBDWT_TBCTRL)

The MTBDWT_TBCTRL register defines how the watchpoint comparisons control the actual trace buffer operation.

Recall the MTB supports starting and stopping the program trace based on the watchpoint comparisons signaled via TSTART and TSTOP. The watchpoint comparison signals are enabled in the MTB's control logic by setting the appropriate enable bits, MTB_MASTER[TSTARTEN, TSTOPEN]. In the event of simultaneous assertion of both TSTART and TSTOP, TSTART takes priority.

Address: F000_1000h base + 200h offset = F000_1200h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														ACOMP1	ACOMP0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MTBDWT_TBCTRL field descriptions

Field	Description
31–28 NUMCOMP	Number of Comparators This read-only field specifies the number of comparators in the MTB_DWT. This implementation includes two registers.
27–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 ACOMP1	Action based on Comparator 1 match When the MTBDWT_FCT1[MATCHED] is set, it indicates MTBDWT_COMP1 address compare has triggered and the trace buffer's recording state is changed. 0 Trigger TSTOP based on the assertion of MTBDWT_FCT1[MATCHED]. 1 Trigger TSTART based on the assertion of MTBDWT_FCT1[MATCHED].
0 ACOMP0	Action based on Comparator 0 match When the MTBDWT_FCT0[MATCHED] is set, it indicates MTBDWT_COMP0 address compare has triggered and the trace buffer's recording state is changed. The assertion of MTBDWT_FCT0[MATCHED] is caused by the following conditions: • Address match in MTBDWT_COMP0 when MTBDWT_FCT0[DATAVMATCH] = 0 • Data match in MTBDWT_COMP0 when MTBDWT_FCT0[DATAVMATCH, DATAVADDR0] = {1,0} • Data match in MTBDWT_COMP0 and address match in MTBDWT_COMP1 when MTBDWT_FCT0[DATAVMATCH, DATAVADDR0] = {1,1} 0 Trigger TSTOP based on the assertion of MTBDWT_FCT0[MATCHED]. 1 Trigger TSTART based on the assertion of MTBDWT_FCT0[MATCHED].

15.32.7 Device Configuration Register (MTBDWT_DEVICECFG)

This register indicates the device configuration. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_1000h base + FC8h offset = F000_1FC8h

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

MTBDWT_DEVICECFG field descriptions

Field	Description
31–0 DEVICECFG	Hardwired to 0x0000_0000.

15.32.8 Device Type Identifier Register (MTBDWT_DEVICETYPID)

This register indicates the device type ID. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_1000h base + FCCh offset = F000_1FCCh

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

MTBDWT_DEVICETYPID field descriptions

Field	Description
31–0 DEVICETYPID	Hardwired to 0x0000_0004.

15.32.9 Peripheral ID Register (MTBDWT_PERIPHID*n*)

These registers indicate the peripheral IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_1000h base + FD0h offset + (4d × i), where i=0d to 7d

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

* Notes:
- See field descriptions for the reset values. x = Undefined at reset.

MTBDWT_PERIPHID*n* field descriptions

Field	Description
31–0 PERIPHID	Peripheral ID1 is hardwired to 0x0000_00E0; ID2 to 0x0000_0008; and all the others to 0x0000_0000.

15.32.10 Component ID Register (MTBDWT_COMPID*n*)

These registers indicate the component IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_1000h base + FF0h offset + (4d × i), where i=0d to 3d

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

* Notes:
- See field descriptions for the reset values.x = Undefined at reset.

MTBDWT_COMPID*n* field descriptions

Field	Description
31–0 COMPID	Component ID Component ID0 is hardwired to 0x0000_000D; ID1 to 0x0000_0090; ID2 to 0x0000_0005; ID3 to 0x0000_00B1.

15.3.3 System ROM Memory Map

The System ROM Table registers are also mapped into a sparsely-populated 4 KB address space.

For core configurations like that supported by Cortex-M0+, ARM recommends that a debugger identifies and connects to the debug components using the CoreSight debug infrastructure.

ARM recommends that a debugger follows the flow as shown in the following figure to discover the components in the CoreSight debug infrastructure. In this case, a debugger reads the peripheral and component ID registers for each CoreSight component in the CoreSight system.

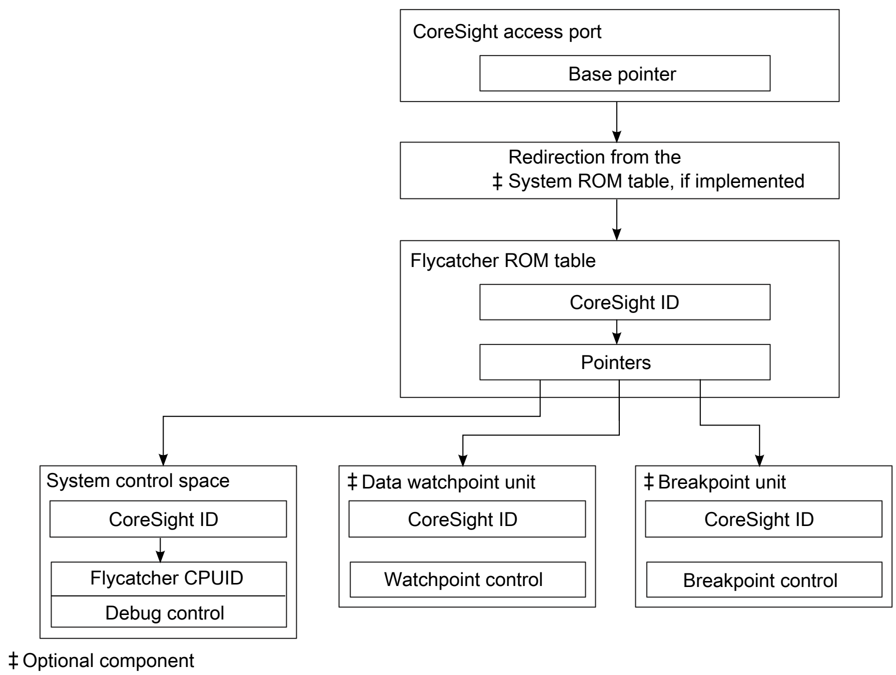


Figure 15-56. CoreSight discovery process

ROM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_2000	Entry (ROM_ENTRY0)	32	R	See section	15.33.1/287
F000_2004	Entry (ROM_ENTRY1)	32	R	See section	15.33.1/287
F000_2008	Entry (ROM_ENTRY2)	32	R	See section	15.33.1/287
F000_200C	End of Table Marker Register (ROM_TABLEMARK)	32	R	0000_0000h	15.33.2/288
F000_2FCC	System Access Register (ROM_SYSACCESS)	32	R	0000_0001h	15.33.3/288
F000_2FD0	Peripheral ID Register (ROM_PERIPHID4)	32	R	See section	15.33.4/289

Table continues on the next page...

ROM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_2FD4	Peripheral ID Register (ROM_PERIPHID5)	32	R	See section	15.33.4/289
F000_2FD8	Peripheral ID Register (ROM_PERIPHID6)	32	R	See section	15.33.4/289
F000_2FDC	Peripheral ID Register (ROM_PERIPHID7)	32	R	See section	15.33.4/289
F000_2FE0	Peripheral ID Register (ROM_PERIPHID0)	32	R	See section	15.33.4/289
F000_2FE4	Peripheral ID Register (ROM_PERIPHID1)	32	R	See section	15.33.4/289
F000_2FE8	Peripheral ID Register (ROM_PERIPHID2)	32	R	See section	15.33.4/289
F000_2FEC	Peripheral ID Register (ROM_PERIPHID3)	32	R	See section	15.33.4/289
F000_2FF0	Component ID Register (ROM_COMPID0)	32	R	See section	15.33.5/289
F000_2FF4	Component ID Register (ROM_COMPID1)	32	R	See section	15.33.5/289
F000_2FF8	Component ID Register (ROM_COMPID2)	32	R	See section	15.33.5/289
F000_2FFC	Component ID Register (ROM_COMPID3)	32	R	See section	15.33.5/289

15.33.1 Entry (ROM_ENTRY*n*)

The System ROM Table begins with "n" relative 32-bit addresses, one for each debug component present in the device and terminating with an all-zero value signaling the end of the table at the "n+1"-th value.

It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_2000h base + 0h offset + (4d × i), where i=0d to 2d

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

* Notes:
- See field descriptions for reset values.x = Undefined at reset.

ROM_ENTRY*n* field descriptions

Field	Description
31–0 ENTRY	ENTRY Entry 0 (MTB) is hardwired to 0xFFFF_E003; Entry 1 (MTBDWT) to 0xFFFF_F003; Entry 2 (CM0+ ROM Table) to 0xF00F_D003.

15.33.2 End of Table Marker Register (ROM_TABLEMARK)

This register indicates end of table marker. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_2000h base + Ch offset = F000_200Ch

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

ROM_TABLEMARK field descriptions

Field	Description
31–0 MARK	Hardwired to 0x0000_0000

15.33.3 System Access Register (ROM_SYSACCESS)

This register indicates system access. It is hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_2000h base + FCCh offset = F000_2FCCh

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

ROM_SYSACCESS field descriptions

Field	Description
31–0 SYSACCESS	Hardwired to 0x0000_0001

15.33.4 Peripheral ID Register (ROM_PERIPHID*n*)

These registers indicate the peripheral IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_2000h base + FD0h offset + (4d × i), where i=0d to 7d

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

* Notes:
- See field descriptions for reset values.x = Undefined at reset.

ROM_PERIPHID*n* field descriptions

Field	Description
31–0 PERIPHID	Peripheral ID1 is hardwired to 0x0000_00E0; ID2 to 0x0000_0008; and all the others to 0x0000_0000.

15.33.5 Component ID Register (ROM_COMPID*n*)

These registers indicate the component IDs. They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Address: F000_2000h base + FF0h offset + (4d × i), where i=0d to 3d

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

* Notes:
- See field descriptions for reset values.x = Undefined at reset.

ROM_COMPID*n* field descriptions

Field	Description
31–0 COMPID	Component ID Component ID0 is hardwired to 0x0000_000D; ID1 to 0x0000_0010; ID2 to 0x0000_0005; ID3 to 0x0000_00B1.

Chapter 16
Crossbar Switch Lite (AXBS-Lite)

16.1 Introduction

NOTE

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

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

The crossbar switch connects bus masters and bus slaves using a crossbar switch structure. This structure allows up to four bus masters to access different bus slaves simultaneously, while providing arbitration among the bus masters when they access the same slave.

16.1.1 Features

The crossbar switch includes these features:

- Symmetric crossbar bus switch implementation
  - Allows concurrent accesses from different masters to different slaves
- 32-bit data bus
- Operation at a 1-to-1 clock frequency with the bus masters
- Programmable configuration for fixed-priority or round-robin slave port arbitration

16.2 Memory Map / Register Definition

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

16.3 Functional Description

16.3.1 General operation

When a master accesses the crossbar switch, the access is immediately taken. If the targeted slave port of the access is available, then the access is immediately presented on the slave port. Single-clock or zero-wait-state accesses are possible through the crossbar. If the targeted slave port of the access is busy or parked on a different master port, the requesting master simply sees wait states inserted until the targeted slave port can service the master's request. The latency in servicing the request depends on each master's priority level and the responding slave's access time.

Because the crossbar switch appears to be just another slave to the master device, the master device has no knowledge of whether it actually owns the slave port it is targeting. While the master does not have control of the slave port it is targeting, it simply waits.

A master is given control of the targeted slave port only after a previous access to a different slave port completes, regardless of its priority on the newly targeted slave port. This prevents deadlock from occurring when:

- A higher priority master has:
  - An outstanding request to one slave port that has a long response time and
  - A pending access to a different slave port, and
- A lower priority master is also making a request to the same slave port as the pending access of the higher priority master.

After the master has control of the slave port it is targeting, the master remains in control of the slave port until it relinquishes the slave port by running an IDLE cycle or by targeting a different slave port for its next access.

The master can also lose control of the slave port if another higher-priority master makes a request to the slave port.

The crossbar terminates all master IDLE transfers, as opposed to allowing the termination to come from one of the slave buses. Additionally, when no master is requesting access to a slave port, the crossbar drives IDLE transfers onto the slave bus, even though a default master may be granted access to the slave port.

When a slave bus is being idled by the crossbar, it remains parked with the last master to use the slave port. This is done to save the initial clock of arbitration delay that otherwise would be seen if the master had to arbitrate to gain control of the slave port.

16.3.2 Arbitration

The crossbar switch supports two arbitration algorithms:

- Fixed priority
- Round-robin

The selection of the global slave port arbitration is controlled by MCM_PLACR[ARB]. For fixed priority, set MCM_PLACR[ARB] to 0. For round robin, set MCM_PLACR[ARB] to 1. This arbitration setting applies to all slave ports.

16.3.2.1 Arbitration during undefined length bursts

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

16.3.2.2 Fixed-priority operation

When operating in fixed-priority mode, each master is assigned a unique priority level with the highest numbered master having the highest priority (for example, in a system with 5 masters, master 1 has lower priority than master 3). If two masters request access to the same slave port, the master with the highest priority gains control over the slave port.

NOTE

In this arbitration mode, a higher-priority master can monopolize a slave port, preventing accesses from any lower-priority master to the port.

When a master makes a request to a slave port, the slave port checks whether the new requesting master's priority level is higher than that of the master that currently has control over the slave port, unless the slave port is in a parked state. The slave port performs an arbitration check at every clock edge to ensure that the proper master, if any, has control of the slave port.

The following table describes possible scenarios based on the requesting master port:

Table 16-1. How AXBS grants control of a slave port to a master

When	Then AXBS 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

16.3.2.3 Round-robin priority operation

When operating in round-robin mode, each master is assigned a relative priority based on the master port number. This relative priority is compared to the master port number (ID) of the last master to perform a transfer on the slave bus. The highest priority requesting master becomes owner of the slave bus at the next transfer boundary. Priority is based on how far ahead the ID of the requesting master is to the ID of the last master.

After granted access to a slave port, a master may perform as many transfers as desired to that port until another master makes a request to the same slave port. The next master in line is granted access to the slave port at the next transfer boundary, or possibly on the next clock cycle if the current master has no pending access request.

As an example of arbitration in round-robin mode, assume the crossbar is implemented with master ports 0, 1, 4, and 5. If the last master of the slave port was master 1, and master 0, 4, and 5 make simultaneous requests, they are serviced in the order: 4 then 5 then 0.

The round-robin arbitration mode generally provides a more fair allocation of the available slave-port bandwidth (compared to fixed priority) as the fixed master priority does not affect the master selection.

16.4 Initialization/application information

No initialization is required for the crossbar switch.

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

Chapter 17
Peripheral Bridge (AIPS-Lite)

17.1 Introduction

NOTE

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

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.

17.1.1 Features

Key features of the peripheral bridge are:

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

17.1.2 General operation

The slave devices connected to the peripheral bridge are modules which contain a programming model of control and status registers. The system masters read and write these registers through the peripheral bridge. The peripheral bridge performs a bus protocol conversion of the master transactions and generates the following as inputs to the peripherals:

- Module enables
- Module addresses

- Transfer attributes
- Byte enables
- Write data

The peripheral bridge selects and captures read data from the peripheral interface and returns it to the crossbar switch.

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.

The AIPS-Lite module uses the data width of accessed peripheral to perform proper data byte lane routing; bus decomposition (bus sizing) is performed when the access size is larger than the peripheral's data width.

17.2 Functional description

The peripheral bridge functions as a bus protocol translator between the crossbar switch and the slave peripheral bus.

The peripheral bridge manages all transactions destined for the attached slave devices and generates select signals for modules on the peripheral bus by decoding accesses within the attached address space.

17.2.1 Access support

All combinations of access size and peripheral data port width are supported. An access that is larger than the target peripheral's data width will be decomposed to multiple, smaller accesses. Bus decomposition is terminated by a transfer error caused by an access to an empty register area.

Chapter 18
Direct Memory Access Multiplexer (DMAMUX)

18.1 Introduction

NOTE

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

18.1.1 Overview

The Direct Memory Access Multiplexer (DMAMUX) routes DMA sources, called slots, to any of the four DMA channels. This process is illustrated in the following figure.

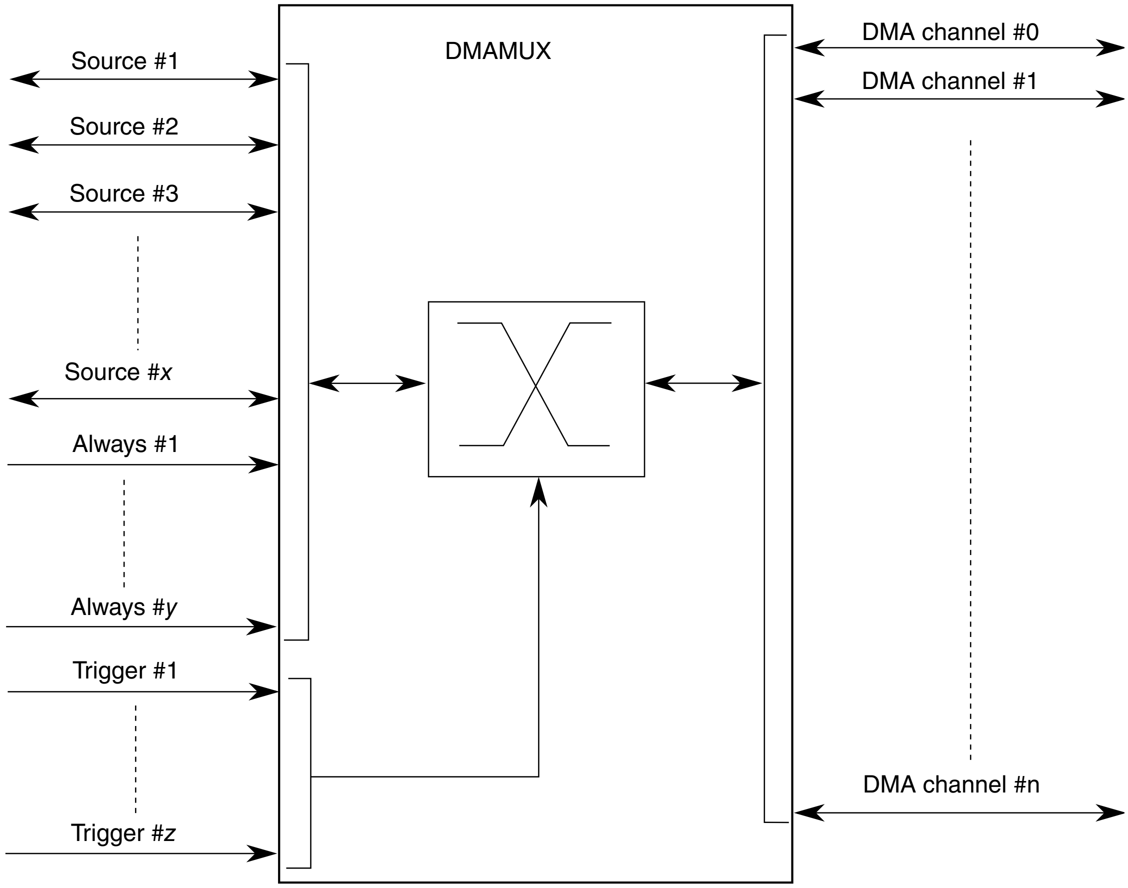


Figure 18-1. DMAMUX block diagram

18.1.2 Features

The DMAMUX module provides these features:

- Up to 63 peripheral slots and up to four always-on slots can be routed to four channels.
- four independently selectable DMA channel routers.
    - The first two channels additionally provide a trigger functionality.
- Each channel router can be assigned to one of the possible peripheral DMA slots or to one of the always-on slots.

18.1.3 Modes of operation

The following operating modes are available:

- Disabled mode

  In this mode, the DMA channel is disabled. Because disabling and enabling of DMA channels is done primarily via the DMA configuration registers, this mode is used mainly as the reset state for a DMA channel in the DMA channel MUX. It may also be used to temporarily suspend a DMA channel while reconfiguration of the system takes place, for example, changing the period of a DMA trigger.

- Normal mode

  In this mode, a DMA source is routed directly to the specified DMA channel. The operation of the DMAMUX in this mode is completely transparent to the system.

- Periodic Trigger mode

  In this mode, a DMA source may only request a DMA transfer, such as when a transmit buffer becomes empty or a receive buffer becomes full, periodically. Configuration of the period is done in the registers of the periodic interrupt timer (PIT). This mode is available only for channels 0–1.

18.2 External signal description

The DMAMUX has no external pins.

18.3 Memory map/register definition

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

DMAMUX memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_1000	Channel Configuration register (DMAMUX0_CHCFG0)	8	R/W	00h	18.3.1/302
4002_1001	Channel Configuration register (DMAMUX0_CHCFG1)	8	R/W	00h	18.3.1/302
4002_1002	Channel Configuration register (DMAMUX0_CHCFG2)	8	R/W	00h	18.3.1/302
4002_1003	Channel Configuration register (DMAMUX0_CHCFG3)	8	R/W	00h	18.3.1/302

18.3.1 Channel Configuration register (DMAMUXx_CHCFGn)

Each of the DMA channels can be independently enabled/disabled and associated with one of the DMA slots (peripheral slots or always-on slots) in the system.

NOTE

Setting multiple CHCFG registers with the same source value will result in unpredictable behavior. This is true, even if a channel is disabled (ENBL==0).

Before changing the trigger or source settings, a DMA channel must be disabled via CHCFGn[ENBL].

Address: 4002_1000h base + 0h offset + (1d × i), where i=0d to 3d

Bit	7	6	5	4	3	2	1	0
Read Write	ENBL	TRIG	SOURCE					
Reset	0	0	0	0	0	0	0	0

DMAMUXx_CHCFGn field descriptions

Field	Description
7 ENBL	DMA Channel Enable Enables the DMA channel. 0 DMA channel is disabled. This mode is primarily used during configuration of the DMAMux. The DMA has separate channel enables/disables, which should be used to disable or reconfigure a DMA channel. 1 DMA channel is enabled
6 TRIG	DMA Channel Trigger Enable Enables the periodic trigger capability for the triggered DMA channel. 0 Triggering is disabled. If triggering is disabled and ENBL is set, the DMA Channel will simply route the specified source to the DMA channel. (Normal mode) 1 Triggering is enabled. If triggering is enabled and ENBL is set, the DMAMUX is in Periodic Trigger mode.
5–0 SOURCE	DMA Channel Source (Slot) Specifies which DMA source, if any, is routed to a particular DMA channel. See your device's chip configuration details for information about the peripherals and their slot numbers.

18.4 Functional description

The primary purpose of the DMAMUX is to provide flexibility in the system's use of the available DMA channels.

As such, configuration of the DMAMUX is intended to be a static procedure done during execution of the system boot code. However, if the procedure outlined in Enabling and configuring sources is followed, the configuration of the DMAMUX may be changed during the normal operation of the system.

Functionally, the DMAMUX channels may be divided into two classes:

- Channels that implement the normal routing functionality plus periodic triggering capability
- Channels that implement only the normal routing functionality

18.4.1 DMA channels with periodic triggering capability

Besides the normal routing functionality, the first 2 channels of the DMAMUX provide a special periodic triggering capability that can be used to provide an automatic mechanism to transmit bytes, frames, or packets at fixed intervals without the need for processor intervention. The trigger is generated by 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.

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.

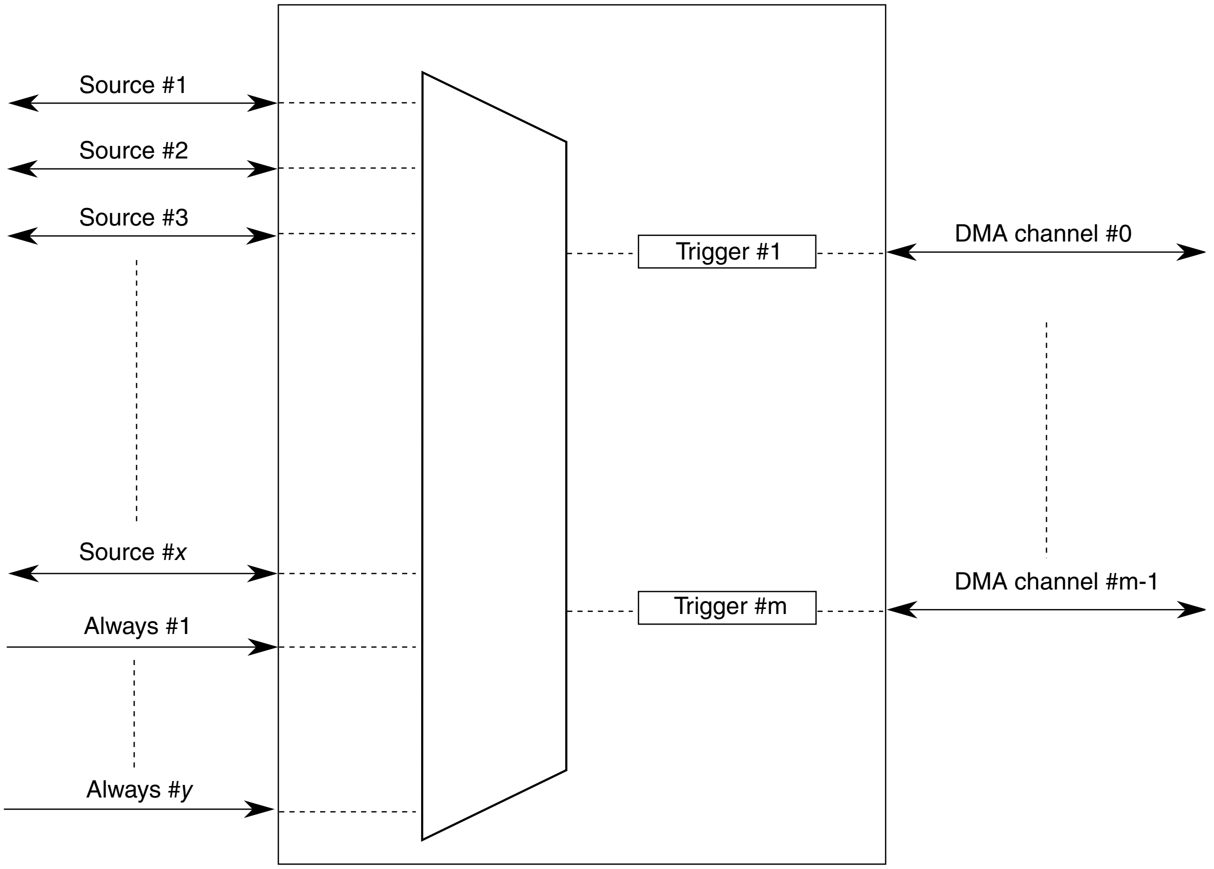


Figure 18-12. DMAMUX triggered channels

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.

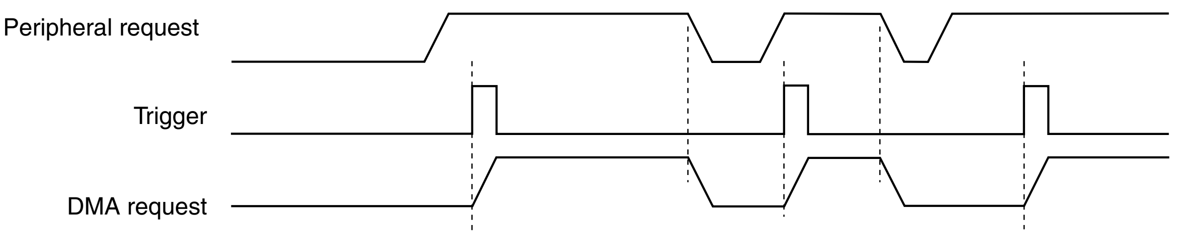


Figure 18-13. DMAMUX channel triggering: normal operation

After the DMA request has been serviced, the peripheral will negate its request, effectively resetting the gating mechanism until the peripheral reasserts its request and the next trigger event is seen. This means that if a trigger is seen, but the peripheral is not requesting a transfer, then that trigger will be ignored. This situation is illustrated in the following figure.

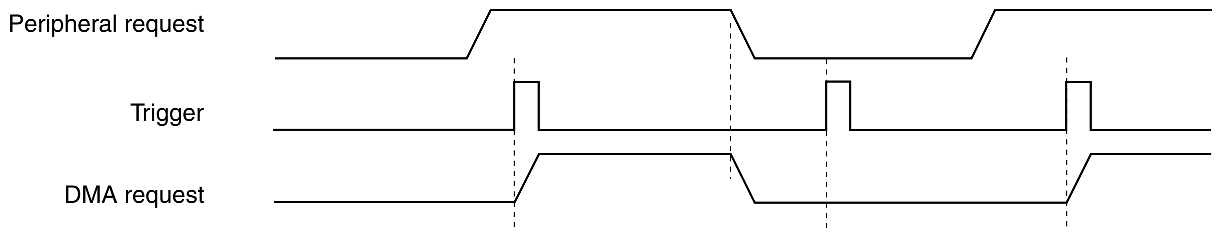


Figure 18-14. DMAMUX channel triggering: ignored trigger

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.

18.4.2 DMA channels with no triggering capability

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

18.4.3 Always-enabled DMA sources

In addition to the peripherals that can be used as DMA sources, there are four additional DMA sources that are always enabled. Unlike the peripheral DMA sources, where the peripheral controls the flow of data during DMA transfers, the sources that are always enabled provide no such "throttling" of the data transfers. These sources are most useful in the following cases:

- Performing DMA transfers to/from GPIO—Moving data from/to one or more GPIO pins, either unthrottled (that is, as fast as possible), or periodically (using the DMA triggering capability).
- Performing DMA transfers from memory to memory—Moving data from memory to memory, typically as fast as possible, sometimes with software activation.
- Performing DMA transfers from memory to the external bus, or vice-versa—Similar to memory to memory transfers, this is typically done as quickly as possible.
- Any DMA transfer that requires software activation—Any DMA transfer that should be explicitly started by software.

In cases where software should initiate the start of a DMA transfer, an always-enabled DMA source can be used to provide maximum flexibility. When activating a DMA channel via software, subsequent executions of the minor loop require that a new start event be sent. This can either be a new software activation, or a transfer request from the DMA channel MUX. The options for doing this are:

- Transfer all data in a single minor loop.

  By configuring the DMA to transfer all of the data in a single minor loop (that is, major loop counter = 1), no reactivation of the channel is necessary. The disadvantage to this option is the reduced granularity in determining the load that the DMA transfer will impose on the system. For this option, the DMA channel must be disabled in the DMA channel MUX.
- Use explicit software reactivation.

  In this option, the DMA is configured to transfer the data using both minor and major loops, but the processor is required to reactivate the channel by writing to the DMA registers *after every minor loop*. For this option, the DMA channel must be disabled in the DMA channel MUX.
- Use an always-enabled DMA source.

In this option, the DMA is configured to transfer the data using both minor and major loops, and the DMA channel MUX does the channel reactivation. For this option, the DMA channel should be enabled and pointing to an "always enabled" source. Note that the reactivation of the channel can be continuous (DMA triggering is disabled) or can use the DMA triggering capability. In this manner, it is possible to execute periodic transfers of packets of data from one source to another, without processor intervention.

18.5 Initialization/application information

This section provides instructions for initializing the DMA channel MUX.

18.5.1 Reset

The reset state of each individual bit is shown in Memory map/register definition. In summary, after reset, all channels are disabled and must be explicitly enabled before use.

18.5.2 Enabling and configuring sources

To enable a source with periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 2 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Configure the corresponding timer.
5. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

NOTE

The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with periodic triggering capability:

1. Write 0x00 to CHCFG1 (base address + 0x01).

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 (base address + 0x01).

The following code example illustrates steps 1 and 4 above:

```
void DMAMUX_Init(uint8_t DMA_CH, uint8_t DMAMUX_SOURCE)
{
    DMAMUX_0.CHCFG[DMA_CH].B.SOURCE = DMAMUX_SOURCE;
    DMAMUX_0.CHCFG[DMA_CH].B.ENBL   = 1;
    DMAMUX_0.CHCFG[DMA_CH].B.TRIG   = 1;
}
```

To enable a source, without periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 2 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that CHCFG[ENBL] is set while CHCFG[TRIG] is cleared.

NOTE

The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with no periodic triggering capability:

1. Write 0x00 to CHCFG1 (base address + 0x01).
2. Configure channel 1 in the DMA, including enabling the channel.
3. Write 0x85 to CHCFG1 (base address + 0x01).

The following code example illustrates steps 1 and 3 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0xFC084000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCFG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCFG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCFG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCFG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCFG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCFG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCFG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCFG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCFG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCFG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCFG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCFG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCFG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCFG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
```

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

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

To disable a source:

A particular DMA source may be disabled by not writing the corresponding source value into any of the CHCFG registers. Additionally, some module-specific configuration may be necessary. See the appropriate section for more details.

To switch the source of a DMA channel:

1. Disable the DMA channel in the DMA and reconfigure the channel for the new source.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] bits of the DMA channel.
3. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

To switch DMA channel 8 from source #5 transmit to source #7 transmit:

1. In the DMA configuration registers, disable DMA channel 8 and reconfigure it to handle the transfers to peripheral slot 7. This example assumes channel 8 doesn't have triggering capability.
2. Write 0x00 to CHCFG8 (base address + 0x08).
3. Write 0x87 to CHCFG8 (base address + 0x08). (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     0xFC084000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCFG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCFG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCFG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCFG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCFG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCFG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCFG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCFG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCFG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCFG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCFG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCFG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCFG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCFG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCFG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCFG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);
```

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

Chapter 19
DMA Controller Module

19.1 Introduction

NOTE

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

Information found here describes the direct memory access (DMA) controller module. It provides an overview of the module and describes in detail its signals and programming model.

The latter sections of this chapter describe operations, features, and supported data transfer modes in detail.

An example of using several features of the DMA module is described in AN4631: Using the Asynchronous DMA features of the Kinetis L Series.

Note

> The designation *n* is used throughout this section to refer to registers or signals associated with one of the four identical DMA channels: DMA0, DMA1, DMA2, or DMA3.

19.1.1 Overview

The DMA controller module enables fast transfers of data, providing an efficient way to move blocks of data with minimal processor interaction. The DMA module, shown in the following figure, has four channels that allow 8-bit, 16-bit, or 32-bit data transfers. Each channel has a dedicated Source Address register (SAR*n*), Destination Address register (DAR*n*), Status register (DSR*n*), Byte Count register (BCR*n*), and Control register (DCR*n*). Collectively, the combined program-visible registers associated with each channel define a transfer control descriptor (TCD). All transfers are dual address, moving data from a source memory location to a destination memory location with the module operating as a 32-bit bus master connected to the system bus. The programming model is accessed through a 32-bit connection with the slave peripheral bus. DMA data transfers may be explicitly initiated by software or by peripheral hardware requests.

The following figure is a simplified block diagram of the 4-channel DMA controller.

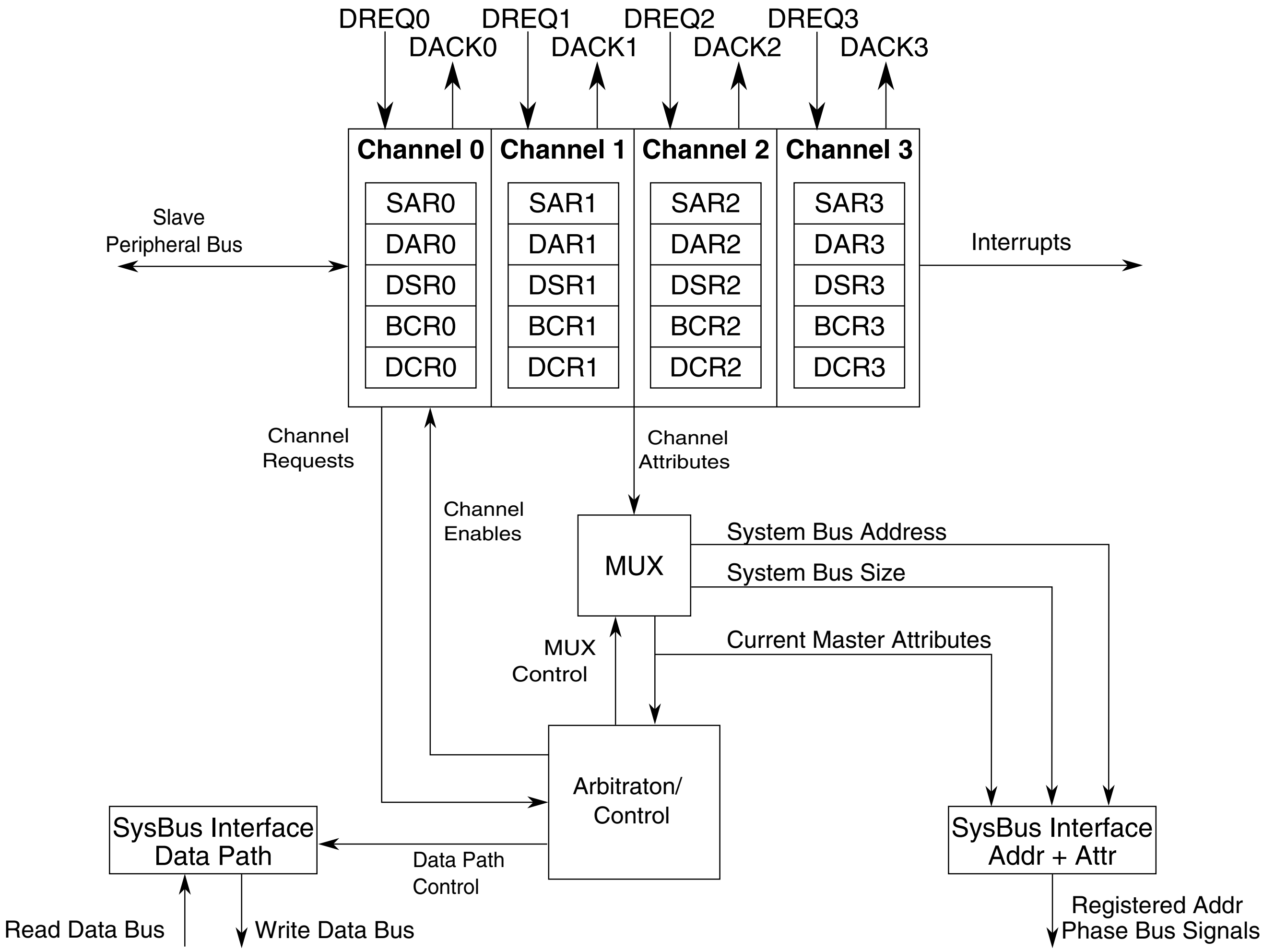


Figure 19-1. 4-Channel DMA Block Diagram

The terms *peripheral request* and *DREQ* refer to a DMA request from one of the on-chip peripherals or package pins. The DMA provides hardware handshake signals: either a DMA acknowledge (DACK) or a done indicator back to the peripheral.

19.1.2 Features

The DMA controller module features:

- Four independently programmable DMA controller channels
- Dual-address transfers via 32-bit master connection to the system bus
- Data transfers in 8-, 16-, or 32-bit blocks
- Continuous-mode or cycle-steal transfers from software or peripheral initiation
- Automatic hardware acknowledge/done indicator from each channel
- Independent source and destination address registers
- Optional modulo addressing and automatic updates of source and destination addresses
- Independent transfer sizes for source and destination
- Optional auto-alignment feature for source or destination accesses
- Optional automatic single or double channel linking
- Programming model accessed via 32-bit slave peripheral bus
- Channel arbitration on transfer boundaries using fixed priority scheme

19.2 DMA Transfer Overview

The DMA module can move data within system memory (including memory and peripheral devices) with minimal processor intervention, greatly improving overall system performance.

The DMA module consists of four independent, functionally equivalent channels, so references to DMA in this chapter apply to any of the channels. It is not possible to address all four channels at once.

As soon as a channel has been initialized, it may be started by setting DCR*n*[START] or a properly-selected peripheral DMA request, depending on the status of DCR*n*[ERQ].

The DMA controller supports dual-address transfers using its bus master connection to the system bus. The DMA channels support transfers up to 32 data bits in size and have the same memory map addressibility as the processor.

- Dual-address transfers—A dual-address transfer consists of a read followed by a write and is initiated by a request using the DCRn[START] bit or by a peripheral DMA request. The read data is temporarily held in the DMA channel hardware until the write operation. Two types of single transfers occur: a read from a source address followed by a write to a destination address. See the following figure.

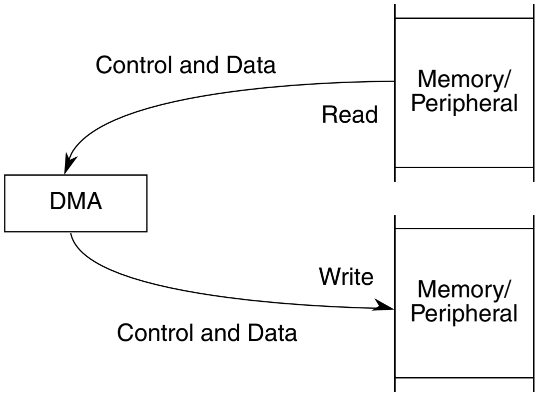


Figure 19-2. Dual-Address Transfer

Any operation involving a DMA channel follows the same three steps:

1. Channel initialization—The transfer control descriptor, contained in the channel registers, is loaded with address pointers, a byte-transfer count, and control information using accesses from the slave peripheral bus.
2. Data transfer—The DMA accepts requests for data transfers. Upon receipt of a request, it provides address and bus control for the transfers via its master connection to the system bus and temporary storage for the read data. The channel performs one or more source read and destination write data transfers.
3. Channel termination—Occurs after the operation is finished successfully or due to an error. The channel indicates the operation status in the channel's DSR, described in the definitions of the DMA Status Registers (DSRn) and Byte Count Registers (BCRn).

19.3 Memory Map/Register Definition

Information about the registers related to the DMA controller module can be found here.

Descriptions of each register and its bit assignments follow. Modifying DMA control registers during a transfer can result in undefined operation. The following table shows the mapping of DMA controller registers. The DMA programming model is accessed via the slave peripheral bus. The concatenation of the source and destination address registers, the status and byte count register, and the control register create a 128-bit transfer control descriptor (TCD) that defines the operation of each DMA channel.

DMA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_8100	Source Address Register (DMA_SAR0)	32	R/W	0000_0000h	19.3.1/316
4000_8104	Destination Address Register (DMA_DAR0)	32	R/W	0000_0000h	19.3.2/317
4000_8108	DMA Status Register / Byte Count Register (DMA_DSR_BCR0)	32	R/W	0000_0000h	19.3.3/317
4000_810C	DMA Control Register (DMA_DCR0)	32	R/W	0000_0000h	19.3.4/320
4000_8110	Source Address Register (DMA_SAR1)	32	R/W	0000_0000h	19.3.1/316
4000_8114	Destination Address Register (DMA_DAR1)	32	R/W	0000_0000h	19.3.2/317
4000_8118	DMA Status Register / Byte Count Register (DMA_DSR_BCR1)	32	R/W	0000_0000h	19.3.3/317
4000_811C	DMA Control Register (DMA_DCR1)	32	R/W	0000_0000h	19.3.4/320
4000_8120	Source Address Register (DMA_SAR2)	32	R/W	0000_0000h	19.3.1/316
4000_8124	Destination Address Register (DMA_DAR2)	32	R/W	0000_0000h	19.3.2/317
4000_8128	DMA Status Register / Byte Count Register (DMA_DSR_BCR2)	32	R/W	0000_0000h	19.3.3/317
4000_812C	DMA Control Register (DMA_DCR2)	32	R/W	0000_0000h	19.3.4/320
4000_8130	Source Address Register (DMA_SAR3)	32	R/W	0000_0000h	19.3.1/316
4000_8134	Destination Address Register (DMA_DAR3)	32	R/W	0000_0000h	19.3.2/317
4000_8138	DMA Status Register / Byte Count Register (DMA_DSR_BCR3)	32	R/W	0000_0000h	19.3.3/317
4000_813C	DMA Control Register (DMA_DCR3)	32	R/W	0000_0000h	19.3.4/320

19.3.1 Source Address Register (DMA_SAR*n*)

Restriction

For this register:

- Only 32-bit writes are allowed. 16-bit and 8-bit writes result in a bus error.
- Only four values are allowed to be written to bits 31-20 of this register. A write of any other value to these bits causes a configuration error when the channel starts to execute. For more information about the configuration error, see the description of the CE field of DSR.

Address: 4000_8000h base + 100h offset + (16d × i), where i=0d to 3d

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

DMA_SAR*n* field descriptions

Field	Description
31–0 SAR	Each SAR contains the byte address used by the DMA controller to read data. The SARn is typically aligned on a 0-modulo-ssize boundary—that is, on the natural alignment of the source data. **Restriction:** Bits 31-20 of this register must be written with one of only four allowed values. Each of these four allowed values corresponds to a valid region of the device's memory map. The allowed values are: • 0x000x_xxxx • 0x1FFx_xxxx • 0x200x_xxxx • 0x400x_xxxx After being written with one of the allowed values, bits 31-20 read back as the written value. After being written with any other value, bits 31-20 read back as an indeterminate value.

19.3.2 Destination Address Register (DMA_DAR*n*)

Restriction

For this register:

- Only 32-bit writes are allowed. 16-bit and 8-bit writes result in a bus error.
- Only four values are allowed to be written to bits 31-20 of this register. A write of any other value to these bits causes a configuration error when the channel starts to execute. For more information about the configuration error, see the description of the CE field of DSR.

Address: 4000_8000h base + 104h offset + (16d × i), where i=0d to 3d

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

DMA_DAR*n* field descriptions

Field	Description
31–0 DAR	Each DAR contains the byte address used by the DMA controller to write data. The DARn is typically aligned on a 0-modulo-dsize boundary—that is, on the natural alignment of the destination data. **Restriction:** Bits 31-20 of this register must be written with one of only four allowed values. Each of these four allowed values corresponds to a valid region of the device's memory map. The allowed values are: • 0x000x_xxxx • 0x1FFx_xxxx • 0x200x_xxxx • 0x400x_xxxx After being written with one of the allowed values, bits 31-20 read back as the written value. After being written with any other value, bits 31-20 read back as an indeterminate value.

19.3.3 DMA Status Register / Byte Count Register (DMA_DSR_BCR*n*)

DSR and BCR are two logical registers that occupy one 32-bit address. DSRn occupies bits 31–24, and BCRn occupies bits 23–0. DSRn contains flags indicating the channel status, and BCRn contains the number of bytes yet to be transferred for a given block.

On the successful completion of the write transfer, BCRn decrements by 1, 2, or 4 for 8-bit, 16-bit, or 32-bit accesses, respectively. BCRn is cleared if a 1 is written to DSR[DONE].

In response to an event, the DMA controller writes to the appropriate DSRn bit. Only a write to DSRn[DONE] results in action. DSRn[DONE] is set when the block transfer is complete.

When a transfer sequence is initiated and BCRn[BCR] is not a multiple of 4 or 2 when the DMA is configured for 32-bit or 16-bit transfers, respectively, DSRn[CE] is set and no transfer occurs.

Address: 4000_8000h base + 108h offset + (16d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	CE	BES	BED	0	REQ	BSY	DONE	BCR							
W								w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

DMA_DSR_BCR*n* field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 CE	Configuration Error Any of the following conditions causes a configuration error: • BCR, SAR, or DAR does not match the requested transfer size. • A value greater than 0F_FFFFh is written to BCR. • Bits 31-20 of SAR or DAR are written with a value other than one of the allowed values. See SAR and DAR. • SSIZE or DSIZE is set to an unsupported value. • BCR equals 0 when the DMA receives a start condition. CE is cleared at hardware reset or by writing a 1 to DONE. 0 No configuration error exists. 1 A configuration error has occurred.

Table continues on the next page...

DMA_DSR_BCR*n* field descriptions (continued)

Field	Description
29 BES	Bus Error on Source BES is cleared at hardware reset or by writing a 1 to DONE. 0 No bus error occurred. 1 The DMA channel terminated with a bus error during the read portion of a transfer.
28 BED	Bus Error on Destination BED is cleared at hardware reset or by writing a 1 to DONE. 0 No bus error occurred. 1 The DMA channel terminated with a bus error during the write portion of a transfer.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 REQ	Request 0 No request is pending or the channel is currently active. Cleared when the channel is selected. 1 The DMA channel has a transfer remaining and the channel is not selected.
25 BSY	Busy 0 DMA channel is inactive. Cleared when the DMA has finished the last transaction. 1 BSY is set the first time the channel is enabled after a transfer is initiated.
24 DONE	Transactions Done Set when all DMA controller transactions complete as determined by transfer count, or based on error conditions. When BCR reaches 0, DONE is set when the final transfer completes successfully. DONE can also be used to abort a transfer by resetting the status bits. When a transfer completes, software must clear DONE before reprogramming the DMA. 0 DMA transfer is not yet complete. Writing a 0 has no effect. 1 DMA transfer completed. Writing a 1 to this bit clears all DMA status bits and should be used in an interrupt service routine to clear the DMA interrupt and error bits.
23–0 BCR	This field contains the number of bytes yet to be transferred for a given block. **Restriction:** BCR must be written with a value equal to or less than 0F_FFFFh. After being written with a value in this range, bits 23-20 of BCR read back as 0000b. A write to BCR of a value greater than 0F_FFFFh causes a configuration error when the channel starts to execute. After being written with a value in this range, bits 23-20 of BCR read back as 0001b.

19.3.4 DMA Control Register (DMA_DCR*n*)

Address: 4000_8000h base + 10Ch offset + (16d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EINT	ERQ	CS	AA	0			Reserved	EADREQ	SINC	SSIZE		DINC	DSIZE		0
W																START
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SMOD				DMOD				D_REQ	0	LINKCC		LCH1		LCH2	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_DCR*n* field descriptions

Field	Description
31 EINT	Enable Interrupt on Completion of Transfer Determines whether an interrupt is generated by completing a transfer or by the occurrence of an error condition. 0 No interrupt is generated. 1 Interrupt signal is enabled.
30 ERQ	Enable Peripheral Request **CAUTION:** Be careful: a collision can occur between START and D_REQ when ERQ is 1. 0 Peripheral request is ignored. 1 Enables peripheral request to initiate transfer. A software-initiated request (setting START) is always enabled.
29 CS	Cycle Steal 0 DMA continuously makes read/write transfers until the BCR decrements to 0. 1 Forces a single read/write transfer per request.
28 AA	Auto-align

Table continues on the next page...

DMA_DCR*n* field descriptions (continued)

Field	Description
	AA and SIZE bits determine whether the source or destination is auto-aligned; that is, transfers are optimized based on the address and size. 0 Auto-align disabled 1 If SSIZE indicates a transfer no smaller than DSIZE, source accesses are auto-aligned; otherwise, destination accesses are auto-aligned. Source alignment takes precedence over destination alignment. If auto-alignment is enabled, the appropriate address register increments, regardless of DINC or SINC.
27–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 Reserved	This field is reserved. **CAUTION:** Must be written as zero; otherwise, undefined behavior results.
23 EADREQ	Enable asynchronous DMA requests Enables the channel to support asynchronous DREQs while the MCU is in Stop mode. 0 Disabled 1 Enabled
22 SINC	Source Increment Controls whether the source address increments after each successful transfer. 0 No change to SAR after a successful transfer. 1 The SAR increments by 1, 2, 4 as determined by the transfer size.
21–20 SSIZE	Source Size Determines the data size of the source bus cycle for the DMA controller. 00 32-bit 01 8-bit 10 16-bit 11 Reserved (generates a configuration error (DSRn[CE]) if incorrectly specified at time of channel activation)
19 DINC	Destination Increment Controls whether the destination address increments after each successful transfer. 0 No change to the DAR after a successful transfer. 1 The DAR increments by 1, 2, 4 depending upon the size of the transfer.
18–17 DSIZE	Destination Size Determines the data size of the destination bus cycle for the DMA controller. 00 32-bit 01 8-bit 10 16-bit 11 Reserved (generates a configuration error (DSRn[CE]) if incorrectly specified at time of channel activation)
16 START	Start Transfer

Table continues on the next page...

DMA_DCR*n* field descriptions (continued)

Field	Description
	0 DMA inactive 1 The DMA begins the transfer in accordance to the values in the TCDn. START is cleared automatically after one module clock and always reads as logic 0.
15–12 SMOD	Source Address Modulo Defines the size of the source data circular buffer used by the DMA Controller. If enabled (SMOD is non-zero), the buffer base address is located on a boundary of the buffer size. The value of this boundary is based upon the initial source address (SAR). The base address should be aligned to a 0-modulo-(circular buffer size) boundary. Misaligned buffers are not possible. The boundary is forced to the value determined by the upper address bits in the field selection. 0000 Buffer disabled 0001 Circular buffer size is 16 bytes. 0010 Circular buffer size is 32 bytes. 0011 Circular buffer size is 64 bytes. 0100 Circular buffer size is 128 bytes. 0101 Circular buffer size is 256 bytes. 0110 Circular buffer size is 512 bytes. 0111 Circular buffer size is 1 KB. 1000 Circular buffer size is 2 KB. 1001 Circular buffer size is 4 KB. 1010 Circular buffer size is 8 KB. 1011 Circular buffer size is 16 KB. 1100 Circular buffer size is 32 KB. 1101 Circular buffer size is 64 KB. 1110 Circular buffer size is 128 KB. 1111 Circular buffer size is 256 KB.
11–8 DMOD	Destination Address Modulo Defines the size of the destination data circular buffer used by the DMA Controller. If enabled (DMOD value is non-zero), the buffer base address is located on a boundary of the buffer size. The value of this boundary depends on the initial destination address (DAR). The base address should be aligned to a 0-modulo-(circular buffer size) boundary. Misaligned buffers are not possible. The boundary is forced to the value determined by the upper address bits in the field selection. 0000 Buffer disabled 0001 Circular buffer size is 16 bytes 0010 Circular buffer size is 32 bytes 0011 Circular buffer size is 64 bytes 0100 Circular buffer size is 128 bytes 0101 Circular buffer size is 256 bytes 0110 Circular buffer size is 512 bytes 0111 Circular buffer size is 1 KB 1000 Circular buffer size is 2 KB 1001 Circular buffer size is 4 KB 1010 Circular buffer size is 8 KB 1011 Circular buffer size is 16 KB 1100 Circular buffer size is 32 KB 1101 Circular buffer size is 64 KB

Table continues on the next page...

DMA_DCR*n* field descriptions (continued)

Field	Description
	1110 Circular buffer size is 128 KB 1111 Circular buffer size is 256 KB
7 D_REQ	Disable Request DMA hardware automatically clears the corresponding DCRn[ERQ] bit when the byte count register reaches 0. 0 ERQ bit is not affected. 1 ERQ bit is cleared when the BCR is exhausted.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 LINKCC	Link Channel Control Allows DMA channels to have their transfers linked. The current DMA channel triggers a DMA request to the linked channels (LCH1 or LCH2) depending on the condition described by the LINKCC bits. If not in cycle steal mode (DCRn[CS]=0) and LINKCC equals 01 or 10, no link to LCH1 occurs. If LINKCC equals 01, a link to LCH1 is created after each cycle-steal transfer performed by the current DMA channel is completed. As the last cycle-steal is performed and the BCR reaches zero, then the link to LCH1 is closed and a link to LCH2 is created. 00 No channel-to-channel linking 01 Perform a link to channel LCH1 after each cycle-steal transfer followed by a link to LCH2 after the BCR decrements to 0. 10 Perform a link to channel LCH1 after each cycle-steal transfer 11 Perform a link to channel LCH1 after the BCR decrements to 0.
3–2 LCH1	Link Channel 1 Indicates the DMA channel assigned as link channel 1. The link channel number cannot be the same as the currently executing channel, and generates a configuration error if this is attempted (DSRn[CE] is set). 00 DMA Channel 0 01 DMA Channel 1 10 DMA Channel 2 11 DMA Channel 3
1–0 LCH2	Link Channel 2 Indicates the DMA channel assigned as link channel 2. The link channel number cannot be the same as the currently executing channel, and generates a configuration error if this is attempted (DSRn[CE] is set). 00 DMA Channel 0 01 DMA Channel 1 10 DMA Channel 2 11 DMA Channel 3

19.4 Functional Description

In the following discussion, the term DMA request implies that DCR*n*[START] is set, or DCRn[ERQ] is set and then followed by assertion of the properly selected DMA peripheral request. DCR*n*[START] is cleared when the channel is activated.

Before initiating a dual-address access, the DMA module verifies that DCR*n*[SSIZE] and DCR*n*[DSIZE] are consistent with the source and destination addresses. If they are not consistent, the configuration error bit, DSR*n*[CE], is set. If misalignment is detected, no transfer occurs, DSR*n*[CE] is set, and, depending on the DCR configuration, an interrupt event may be issued. If the auto-align bit, DCR*n*[AA], is set, error checking is performed on the appropriate registers.

A read/write transfer sequence reads data from the source address and writes it to the destination address. The number of bytes transferred is the largest of the sizes specified by DCR*n*[SSIZE] and DCR*n*[DSIZE] in the DMA Control Registers (DCRn).

Source and destination address registers (SAR*n* and DAR*n*) can be programmed in the DCR*n* to increment at the completion of a successful transfer.

19.4.1 Transfer requests (Cycle-Steal and Continuous modes)

The DMA channel supports software-initiated or peripheral-initiated requests. A request is issued by setting DCR*n*[START] or when the selected peripheral request asserts and DCRn[ERQ] is set. Setting DCR*n*[ERQ] enables recognition of the peripheral DMA requests. Selecting between cycle-steal and continuous modes minimizes bus usage for either type of request.

- Cycle-steal mode (DCR*n*[CS] = 1)—Only one complete transfer from source to destination occurs for each request. If DCRn[ERQ] is set, the request is peripheral initiated. A software-initiated request is enabled by setting DCR*n*[START].
- Continuous mode (DCR*n*[CS] = 0)—After a software-initiated or peripheral request, the DMA continuously transfers data until BCRn reaches 0. The DMA performs the specified number of transfers, then retires the channel.

In either mode, the crossbar switch performs independent arbitration on each slave port after each transaction.

19.4.2 Channel initialization and startup

Before a data transfer starts, the channel's transfer control descriptor must be initialized with information describing configuration, request-generation method, and pointers to the data to be moved.

19.4.2.1 Channel prioritization

The four DMA channels are prioritized based on number, with channel 0 having highest priority and channel 3 having the lowest, that is, channel 0 > channel 1 > channel 2 > channel 3.

Simultaneous peripheral requests activate the channels based on this priority order. Once activated, a channel runs to completion as defined by DCR*n*[CS] and BCR*n*.

19.4.2.2 Programming the DMA Controller Module

CAUTION

During a channel's execution, writes to programming model registers can corrupt the data transfer. The DMA module itself does not have a mechanism to prevent writes to registers during a channel's execution.

General guidelines for programming the DMA are:

- TCD*n* is initialized.
  - SAR*n* is loaded with the source (read) address. If the transfer is from a peripheral device to memory or to another peripheral, the source address is the location of the peripheral data register. If the transfer is from memory to a peripheral device or to memory, the source address is the starting address of the data block. This can be any appropriately aligned address.
  - DAR*n* is initialized with the destination (write) address. If the transfer is from a peripheral device to memory, or from memory to memory, DAR*n* is loaded with the starting address of the data block to be written. If the transfer is from memory to a peripheral device, or from a peripheral device to a peripheral device, DAR*n* is loaded with the address of the peripheral data register. This address can be any appropriately aligned address.

- SARn and DARn change after each data transfer depending on DCRn[SSIZE, DSIZE, SINC, DINC, SMOD, DMOD] and the starting addresses. Increment values can be 1, 2, or 4 for 8-bit, 16-bit, or 32-bit transfers, respectively. If the address register is programmed to remain unchanged, the register is not incremented after the data transfer.
- BCRn[BCR] must be loaded with the total number of bytes to be transferred. It is decremented by 1, 2, or 4 at the end of each transfer, depending on the transfer size. DSRn[DONE] must be cleared for channel startup.

- After the channel has been initialized, it may be started by setting DCRn[START] or a properly selected peripheral DMA request, depending on the status of DCRn[ERQ]. For a software-initiated transfer, the channel can be started by setting DCRn[START] as part of a single 32-bit write to the last 32 bits of the TCDn; that is, it is not required to write the DCRn with START cleared and then perform a second write to explicitly set START.
- Programming the channel for a software-initiated request causes the channel to request the system bus and start transferring data immediately. If the channel is programmed for peripheral-initiated request, a properly selected peripheral DMA request must be asserted before the channel begins the system bus transfers.
- The hardware can automatically clear DCRn[ERQ], disabling the peripheral request, when BCRn reaches zero by setting DCRn[D_REQ].
- Changes to DCRn are effective immediately while the channel is active. To avoid problems with changing a DMA channel setup, write a one to DSRn[DONE] to stop the DMA channel.

19.4.3 Dual-Address Data Transfer Mode

Each channel supports dual-address transfers. Dual-address transfers consist of a source data read and a destination data write. The DMA controller module begins a dual-address transfer sequence after a DMA request. If no error condition exists, DSRn[REQ] is set.

- Dual-address read—The DMA controller drives the SARn value onto the system address bus. If DCRn[SINC] is set, the SARn increments by the appropriate number of bytes upon a successful read cycle. When the appropriate number of read cycles complete (multiple reads if the destination size is larger than the source), the DMA initiates the write portion of the transfer.

If a termination error occurs, DSRn[BES, DONE] are set and DMA transactions stop.

- Dual-address write—The DMA controller drives the DAR*n* value onto the system address bus. When the appropriate number of write cycles complete (multiple writes if the source size is larger than the destination), DAR*n* increments by the appropriate number of bytes if DCRn[DINC] is set. BCRn decrements by the appropriate number of bytes. DSR*n*[DONE] is set when BCR*n* reaches zero. If the BCR*n* is greater than zero, another read/write transfer is initiated if continuous mode is enabled (DCRn[CS] = 0).

  If a termination error occurs, DSR*n*[BED, DONE] are set and DMA transactions stop.

19.4.4 Advanced Data Transfer Controls: Auto-Alignment

Typically, auto-alignment for DMA transfers applies for transfers of large blocks of data. As a result, it does not apply for peripheral-initiated cycle-steal transfers.

Auto-alignment allows block transfers to occur at the optimal size based on the address, byte count, and programmed size. To use this feature, DCR*n*[AA] must be set. The source is auto-aligned if DCR*n*[SSIZE] indicates a transfer size larger than DCR*n*[DSIZE]. Source alignment takes precedence over the destination when the source and destination sizes are equal. Otherwise, the destination is auto-aligned. The address register chosen for alignment increments regardless of the increment value. Configuration error checking is performed on registers not chosen for alignment.

If BCR*n* is greater than 16, the address determines transfer size. Transfers of 8 bits, 16 bits, or 32 bits are transferred until the address is aligned to the programmed size boundary, at which time accesses begin using the programmed size. If BCR*n* is less than 16 at the start of a transfer, the number of bytes remaining dictates transfer size.

Consider this example:

- AA equals 1.
- SAR*n* equals 0x2000_0001.
- BCR*n* equals 0x00_00F0.
- SSIZE equals 00 (32 bits).
- DSIZE equals 01 (8 bits).

Because SSIZE > DSIZE, the source is auto-aligned. Error checking is performed on destination registers. The access sequence is as follows:

1. Read 1 byte from 0x2000_0001, increment SAR*n*, write 1 byte (using DAR*n*).
2. Read 2 bytes from 0x2000_0002, increment SAR*n*, write 2 bytes.

3. Read 4 bytes from 0x2000_0004, increment SAR*n*, write 4 bytes.
4. Repeat 4-byte operations until SAR*n* equals 0x2000_00F0.
5. Read byte from 0x2000_00F0, increment SAR*n*, write byte.

If DSIZE is another size, data writes are optimized to write the largest size allowed based on the address, but not exceeding the configured size.

19.4.5 Termination

An unsuccessful transfer can terminate for one of the following reasons:

- Error conditions—When the DMA encounters a read or write cycle that terminates with an error condition, DSR*n*[BES] is set for a read and DSR*n*[BED] is set for a write before the transfer is halted. If the error occurred in a write cycle, data in the internal holding registers is lost.
- Interrupts—If DCR*n*[EINT] is set, the DMA drives the appropriate interrupt request signal. The processor can read DSR*n* to determine whether the transfer terminated successfully or with an error. DSR*n*[DONE] is then written with a 1 to clear the interrupt, DSR*n*[DONE], and error status bits.

Chapter 20
Multipurpose Clock Generator (MCG)

20.1 Introduction

NOTE

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

The multipurpose clock generator (MCG) module provides several clock source choices for the MCU.

The module contains a frequency-locked loop (FLL) and a phase-locked loop (PLL). The FLL is controllable by either an internal or an external reference clock. The PLL is controllable by the external reference clock. The module can select either an FLL or PLL output clock, or a reference clock (internal or external) as a source for the MCU system clock. The MCG operates in conjuction with a crystal oscillator, which allows an external crystal, ceramic resonator, or another external clock source to produce the external reference clock.

20.1.1 Features

Key features of the MCG module are:

- Frequency-locked loop (FLL):
    - Digitally-controlled oscillator (DCO)
    - DCO frequency range is programmable for up to four different frequency ranges.
    - Option to program and maximize DCO output frequency for a low frequency external reference clock source.
    - Option to prevent FLL from resetting its current locked frequency when switching clock modes if FLL reference frequency is not changed.

- Internal or external reference clock can be used as the FLL source.
- Can be used as a clock source for other on-chip peripherals.
- Phase-locked loop (PLL):
  - Voltage-controlled oscillator (VCO)
  - External reference clock is used as the PLL source.
  - Modulo VCO frequency divider
  - Phase/Frequency detector
  - Integrated loop filter
  - Can be used as a clock source for other on-chip peripherals.
- Internal reference clock generator:
  - Slow clock with nine trim bits for accuracy
  - Fast clock with four trim bits
  - Can be used as source clock for the FLL. In FEI mode, only the slow Internal Reference Clock (IRC) can be used as the FLL source.
  - Either the slow or the fast clock can be selected as the clock source for the MCU.
  - Can be used as a clock source for other on-chip peripherals.
- Control signals for the MCG external reference low power oscillator clock generators are provided:
  - HGO0, RANGE0, EREFS0

    HGO1, RANGE1, EREFS1
- External clock from the Crystal Oscillator (OSC0):
  - Can be used as a source for the FLL and/or the PLL.
  - Can be selected as the clock source for the MCU.
- External clock from the Real Time Counter (RTC):
  - Can only be used as a source for the FLL.
  - Can be selected as the clock source for the MCU.
- External clock from the Crystal Oscillator (OSC1)
  - Can only be used as a source for the PLL.

- External clock monitor with reset and interrupt request capability to check for external clock failure when running in FBE, PEE, BLPE, or FEE modes
- Lock detector with interrupt request capability for use with the PLL
- Internal Reference Clocks Auto Trim Machine (ATM) capability using an external clock as a reference
- Reference dividers for both the FLL and the PLL are provided
- Reference dividers for the Fast Internal Reference Clock are provided
- MCG PLL Clock (MCGPLLCLK) is provided as a clock source for other on-chip peripherals
- MCG FLL Clock (MCGFLLCLK) is provided as a clock source for other on-chip peripherals
- MCG Fixed Frequency Clock (MCGFFCLK) is provided as a clock source for other on-chip peripherals
- MCG Internal Reference Clock (MCGIRCLK) is provided as a clock source for other on-chip peripherals

This figure presents the block diagram of the MCG module.

Figure 20-1. Multipurpose Clock Generator (MCG) block diagram

NOTE

Refer to the chip configuration chapter to identify the oscillator used in this MCU.

20.1.2 Modes of Operation

The MCG has the following modes of operation: FEI, FEE, FBI, FBE, PBE, PEE, BLPI, BLPE, and Stop. For details, see MCG modes of operation.

20.2 External Signal Description

There are no MCG signals that connect off chip.

20.3 Memory Map/Register Definition

This section includes the memory map and register definition.

The MCG registers can only be written when in supervisor mode. Write accesses when in user mode will result in a bus error. Read accesses may be performed in both supervisor and user mode.

MCG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_4000	MCG Control 1 Register (MCG_C1)	8	R/W	04h	20.3.1/332
4006_4001	MCG Control 2 Register (MCG_C2)	8	R/W	See section	20.3.2/334
4006_4002	MCG Control 3 Register (MCG_C3)	8	R/W	Undefined	20.3.3/335
4006_4003	MCG Control 4 Register (MCG_C4)	8	R/W	Undefined	20.3.4/336
4006_4004	MCG Control 5 Register (MCG_C5)	8	R/W	00h	20.3.5/337
4006_4005	MCG Control 6 Register (MCG_C6)	8	R/W	00h	20.3.6/338
4006_4006	MCG Status Register (MCG_S)	8	R	10h	20.3.7/340
4006_4008	MCG Status and Control Register (MCG_SC)	8	R/W	02h	20.3.8/341
4006_400A	MCG Auto Trim Compare Value High Register (MCG_ATCVH)	8	R/W	00h	20.3.9/343
4006_400B	MCG Auto Trim Compare Value Low Register (MCG_ATCVL)	8	R/W	00h	20.3.10/343
4006_400C	MCG Control 7 Register (MCG_C7)	8	R/W	00h	20.3.11/343
4006_400D	MCG Control 8 Register (MCG_C8)	8	R/W	80h	20.3.12/344
4006_400F	MCG Control 10 Register (MCG_C10)	8	R/W	00h	20.3.13/345

20.3.1 MCG Control 1 Register (MCG_C1)

Address: 4006_4000h base + 0h offset = 4006_4000h

Bit	7	6	5	4	3	2	1	0
Read / Write	CLKS		FRDIV			IREFS	IRCLKEN	IREFSTEN
Reset	0	0	0	0	0	1	0	0

MCG_C1 field descriptions

Field	Description
7–6 CLKS	Clock Source Select Selects the clock source for MCGOUTCLK . 00 Encoding 0 — Output of FLL or PLL is selected (depends on PLLS control bit).

Table continues on the next page...

MCG_C1 field descriptions (continued)

Field	Description
	01 Encoding 1 — Internal reference clock is selected. 10 Encoding 2 — External reference clock is selected. 11 Encoding 3 — Reserved.
5–3 FRDIV	FLL External Reference Divider Selects the amount to divide down the external reference clock for the FLL. The resulting frequency must be in the range 31.25 kHz to 39.0625 kHz (This is required when FLL/DCO is the clock source for MCGOUTCLK . In FBE mode, it is not required to meet this range, but it is recommended in the cases when trying to enter a FLL mode from FBE). 000 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 1; for all other RANGE 0 values, Divide Factor is 32. 001 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 2; for all other RANGE 0 values, Divide Factor is 64. 010 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 4; for all other RANGE 0 values, Divide Factor is 128. 011 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 8; for all other RANGE 0 values, Divide Factor is 256. 100 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 16; for all other RANGE 0 values, Divide Factor is 512. 101 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 32; for all other RANGE 0 values, Divide Factor is 1024. 110 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 64; for all other RANGE 0 values, Divide Factor is 1280 . 111 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 128; for all other RANGE 0 values, Divide Factor is 1536 .
2 IREFS	Internal Reference Select Selects the reference clock source for the FLL. 0 External reference clock is selected. 1 The slow internal reference clock is selected.
1 IRCLKEN	Internal Reference Clock Enable Enables the internal reference clock for use as MCGIRCLK. 0 MCGIRCLK inactive. 1 MCGIRCLK active.
0 IREFSTEN	Internal Reference Stop Enable Controls whether or not the internal reference clock remains enabled when the MCG enters Stop mode. 0 Internal reference clock is disabled in Stop mode. 1 Internal reference clock is enabled in Stop mode if IRCLKEN is set or if MCG is in FEI, FBI, or BLPI modes before entering Stop mode.

20.3.2 MCG Control 2 Register (MCG_C2)

Address: 4006_4000h base + 1h offset = 4006_4001h

Bit	7	6	5	4	3	2	1	0
Read / Write	LOCRE0	FCFTRIM	RANGE0		HGO0	EREFS0	LP	IRCS
Reset	1	1	0	0	0	0	0	0

MCG_C2 field descriptions

Field	Description
7 LOCRE0	Loss of Clock Reset Enable Determines whether an interrupt or a reset request is made following a loss of OSC0 external reference clock. The LOCRE0 only has an affect when CME0 is set. 0 Interrupt request is generated on a loss of OSC0 external reference clock. 1 Generate a reset request on a loss of OSC0 external reference clock.
6 FCFTRIM	Fast Internal Reference Clock Fine Trim FCFTRIM controls the smallest adjustment of the fast internal reference clock frequency. Setting FCFTRIM increases the period and clearing FCFTRIM decreases the period by the smallest amount possible. If an FCFTRIM value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this bit.
5–4 RANGE0	Frequency Range Select Selects the frequency range for the crystal oscillator or external clock source. See the Oscillator (OSC) chapter for more details and the device data sheet for the frequency ranges used. 00 Encoding 0 — Low frequency range selected for the crystal oscillator . 01 Encoding 1 — High frequency range selected for the crystal oscillator . 1X Encoding 2 — Very high frequency range selected for the crystal oscillator .
3 HGO0	High Gain Oscillator Select Controls the crystal oscillator mode of operation. See the Oscillator (OSC) chapter for more details. 0 Configure crystal oscillator for low-power operation. 1 Configure crystal oscillator for high-gain operation.
2 EREFS0	External Reference Select Selects the source for the external reference clock. See the Oscillator (OSC) chapter for more details. 0 External reference clock requested. 1 Oscillator requested.
1 LP	Low Power Select Controls whether the FLL or PLL is disabled in BLPI and BLPE modes. In FBE or PBE modes, setting this bit to 1 will transition the MCG into BLPE mode; in FBI mode, setting this bit to 1 will transition the MCG into BLPI mode. In any other MCG mode, LP bit has no affect. 0 FLL or PLL is not disabled in bypass modes. 1 FLL or PLL is disabled in bypass modes (lower power)

Table continues on the next page...

MCG_C2 field descriptions (continued)

Field	Description
0 IRCS	Internal Reference Clock Select Selects between the fast or slow internal reference clock source. 0 Slow internal reference clock selected. 1 Fast internal reference clock selected.

20.3.3 MCG Control 3 Register (MCG_C3)

Address: 4006_4000h base + 2h offset = 4006_4002h

Bit	7	6	5	4	3	2	1	0
Read Write	SCTRIM							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

MCG_C3 field descriptions

Field	Description
7–0 SCTRIM	Slow Internal Reference Clock Trim Setting SCTRIM [1] controls the slow internal reference clock frequency by controlling the slow internal reference clock period. The SCTRIM bits are binary weighted, that is, bit 1 adjusts twice as much as bit 0. Increasing the binary value increases the period, and decreasing the value decreases the period. An additional fine trim bit is available in C4 register as the SCFTRIM bit. Upon reset, this value is loaded with a factory trim value. If an SCTRIM value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this register.

1. A value for SCTRIM is loaded during reset from a factory programmed location.

20.3.4 MCG Control 4 Register (MCG_C4)

NOTE

Reset values for DRST and DMX32 bits are 0.

Address: 4006_4000h base + 3h offset = 4006_4003h

Bit	7	6	5	4	3	2	1	0
Read / Write	DMX32	DRST_DRS		FCTRIM				SCFTRIM
Reset	0	0	0	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.
- A value for FCTRIM is loaded during reset from a factory programmed location. x = Undefined at reset.

MCG_C4 field descriptions

Field	Description
7 DMX32	DCO Maximum Frequency with 32.768 kHz Reference The DMX32 bit controls whether the DCO frequency range is narrowed to its maximum frequency with a 32.768 kHz reference. The following table identifies settings for the DCO frequency range. **NOTE:** The system clocks derived from this source should not exceed their specified maximums. (see table below) 0 DCO has a default range of 25%. 1 DCO is fine-tuned for maximum frequency with 32.768 kHz reference.
6–5 DRST_DRS	DCO Range Select The DRS bits select the frequency range for the FLL output, DCOOUT. When the LP bit is set, writes to the DRS bits are ignored. The DRST read field indicates the current frequency range for DCOOUT. The DRST field does not update immediately after a write to the DRS field due to internal synchronization between clock domains. See the DCO Frequency Range table for more details. 00 Encoding 0 — Low range (reset default).

DRST_DRS	DMX32	Reference Range	FLL Factor	DCO Range
00	0	31.25–39.0625 kHz	640	20–25 MHz
	1	32.768 kHz	732	24 MHz
01	0	31.25–39.0625 kHz	1280	40–50 MHz
	1	32.768 kHz	1464	48 MHz
10	0	31.25–39.0625 kHz	1920	60–75 MHz
	1	32.768 kHz	2197	72 MHz
11	0	31.25–39.0625 kHz	2560	80–100 MHz
	1	32.768 kHz	2929	96 MHz

Table continues on the next page...

MCG_C4 field descriptions (continued)

Field	Description
	01 Encoding 1 — Mid range. 10 Encoding 2 — Mid-high range. 11 Encoding 3 — High range.
4–1 FCTRIM	Fast Internal Reference Clock Trim Setting FCTRIM [1] controls the fast internal reference clock frequency by controlling the fast internal reference clock period. The FCTRIM bits are binary weighted, that is, bit 1 adjusts twice as much as bit 0. Increasing the binary value increases the period, and decreasing the value decreases the period. If an FCTRIM[3:0] value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this register.
0 SCFTRIM	Slow Internal Reference Clock Fine Trim SCFTRIM [2] controls the smallest adjustment of the slow internal reference clock frequency. Setting SCFTRIM increases the period and clearing SCFTRIM decreases the period by the smallest amount possible. If an SCFTRIM value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this bit.

1. A value for FCTRIM is loaded during reset from a factory programmed location.
2. A value for SCFTRIM is loaded during reset from a factory programmed location .

20.3.5 MCG Control 5 Register (MCG_C5)

Address: 4006_4000h base + 4h offset = 4006_4004h

Bit	7	6	5	4	3	2	1	0
Read	0	PLLCLKEN0	PLLSTEN0	PRDIV0				
Write								
Reset	0	0	0	0	0	0	0	0

MCG_C5 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 PLLCLKEN0	PLL Clock Enable Enables the PLL independent of PLLS and enables the PLL clock for use as MCGPLLCLK. (PRDIV 0 needs to be programmed to the correct divider to generate a PLL reference clock in the range of 2 - 4 MHz range prior to setting the PLLCLKEN 0 bit). Setting PLLCLKEN 0 will enable the external oscillator if not already enabled. Whenever the PLL is being enabled by means of the PLLCLKEN 0 bit, and the external oscillator is being used as the reference clock, the OSCINIT 0 bit should be checked to make sure it is set. 0 MCGPLLCLK is inactive. 1 MCGPLLCLK is active.
5 PLLSTEN0	PLL Stop Enable

Table continues on the next page...

MCG_C5 field descriptions (continued)

Field	Description
	Enables the PLL Clock during Normal Stop. In Low Power Stop mode, the PLL clock gets disabled even if PLLSTEN 0 =1. All other power modes, PLLSTEN 0 bit has no affect and does not enable the PLL Clock to run if it is written to 1. 0 MCGPLLCLK is disabled in any of the Stop modes. 1 MCGPLLCLK is enabled if system is in Normal Stop mode.
4–0 PRDIV0	PLL External Reference Divider Selects the amount to divide down the external reference clock for the PLL. The resulting frequency must be in the range of 2 MHz to 4 MHz. After the PLL is enabled (by setting either PLLCLKEN 0 or PLLS), the PRDIV 0 value must not be changed when LOCK0 is zero.

Table 20-7. PLL External Reference Divide Factor

PRDIV 0	Divide Factor		PRDIV 0	Divide Factor		PRDIV 0	Divide Factor		PRDIV 0	Divide Factor
00000	1		01000	9		10000	17		11000	25
00001	2		01001	10		10001	18		11001	Reserved
00010	3		01010	11		10010	19		11010	Reserved
00011	4		01011	12		10011	20		11011	Reserved
00100	5		01100	13		10100	21		11100	Reserved
00101	6		01101	14		10101	22		11101	Reserved
00110	7		01110	15		10110	23		11110	Reserved
00111	8		01111	16		10111	24		11111	Reserved

20.3.6 MCG Control 6 Register (MCG_C6)

Address: 4006_4000h base + 5h offset = 4006_4005h

Bit	7	6	5	4	3	2	1	0
Read Write	LOLIE0	PLLS	CME0	VDIV0				
Reset	0	0	0	0	0	0	0	0

MCG_C6 field descriptions

Field	Description
7 LOLIE0	Loss of Lock Interrrupt Enable

Table continues on the next page...

MCG_C6 field descriptions (continued)

Field	Description
	Determines if an interrupt request is made following a loss of lock indication. This bit only has an effect when LOLS 0 is set. 0 No interrupt request is generated on loss of lock. 1 Generate an interrupt request on loss of lock.
6 PLLS	PLL Select Controls whether the PLL or FLL output is selected as the MCG source when CLKS[1:0]=00. If the PLLS bit is cleared and PLLCLKEN 0 is not set, the PLL is disabled in all modes. If the PLLS is set, the FLL is disabled in all modes. 0 FLL is selected. 1 PLL is selected (PRDIV 0 need to be programmed to the correct divider to generate a PLL reference clock in the range of 2–4 MHz prior to setting the PLLS bit).
5 CME0	Clock Monitor Enable Enables the loss of clock monitoring circuit for the OSC0 external reference mux select. The LOCRE0 bit will determine if a interrupt or a reset request is generated following a loss of OSC0 indication. The CME0 bit must only be set to a logic 1 when the MCG is in an operational mode that uses the external clock (FEE, FBE, PEE, PBE, or BLPE) . Whenever the CME0 bit is set to a logic 1, the value of the RANGE0 bits in the C2 register should not be changed. CME0 bit should be set to a logic 0 before the MCG enters any Stop mode. Otherwise, a reset request may occur while in Stop mode. CME0 should also be set to a logic 0 before entering VLPR or VLPW power modes if the MCG is in BLPE mode. 0 External clock monitor is disabled for OSC0. 1 External clock monitor is enabled for OSC0.
4–0 VDIV0	VCO 0 Divider Selects the amount to divide the VCO output of the PLL. The VDIV 0 bits establish the multiplication factor (M) applied to the reference clock frequency. After the PLL is enabled (by setting either PLLCLKEN 0 or PLLS), the VDIV 0 value must not be changed when LOCK 0 is zero.

Table 20-9. PLL VCO Divide Factor

VDIV 0	Multiply Factor		VDIV 0	Multiply Factor		VDIV 0	Multiply Factor		VDIV 0	Multiply Factor
00000	24		01000	32		10000	40		11000	48
00001	25		01001	33		10001	41		11001	49
00010	26		01010	34		10010	42		11010	50
00011	27		01011	35		10011	43		11011	51
00100	28		01100	36		10100	44		11100	52
00101	29		01101	37		10101	45		11101	53
00110	30		01110	38		10110	46		11110	54
00111	31		01111	39		10111	47		11111	55

20.3.7 MCG Status Register (MCG_S)

Address: 4006_4000h base + 6h offset = 4006_4006h

Bit	7	6	5	4	3	2	1	0
Read	LOLS0	LOCK0	PLLST	IREFST	CLKST		OSCINIT0	IRCST
Write								
Reset	0	0	0	1	0	0	0	0

MCG_S field descriptions

Field	Description
7 LOLS0	Loss of Lock Status This bit is a sticky bit indicating the lock status for the PLL. LOLS is set if after acquiring lock, the PLL output frequency has fallen outside the lock exit frequency tolerance, D_{unl} . LOLIE determines whether an interrupt request is made when LOLS is set. LOLRE determines whether a reset request is made when LOLS is set. This bit is cleared by reset or by writing a logic 1 to it when set. Writing a logic 0 to this bit has no effect. 0 PLL has not lost lock since LOLS 0 was last cleared. 1 PLL has lost lock since LOLS 0 was last cleared.
6 LOCK0	Lock Status This bit indicates whether the PLL has acquired lock. Lock detection is only enabled when the PLL is enabled (either through clock mode selection or PLLCLKEN0=1 setting). While the PLL clock is locking to the desired frequency, the MCG PLL clock (MCGPLLCLK) will be gated off until the LOCK bit gets asserted. If the lock status bit is set, changing the value of the PRDIV0 [4:0] bits in the C5 register or the VDIV0[4:0] bits in the C6 register causes the lock status bit to clear and stay cleared until the PLL has reacquired lock. Loss of PLL reference clock will also cause the LOCK0 bit to clear until the PLL has reacquired lock. Entry into LLS, VLPS, or regular Stop with PLLSTEN=0 also causes the lock status bit to clear and stay cleared until the Stop mode is exited and the PLL has reacquired lock. Any time the PLL is enabled and the LOCK0 bit is cleared, the MCGPLLCLK will be gated off until the LOCK0 bit is asserted again. 0 PLL is currently unlocked. 1 PLL is currently locked.
5 PLLST	PLL Select Status This bit indicates the clock source selected by PLLS . The PLLST bit does not update immediately after a write to the PLLS bit due to internal synchronization between clock domains. 0 Source of PLLS clock is FLL clock. 1 Source of PLLS clock is PLL output clock.
4 IREFST	Internal Reference Status This bit indicates the current source for the FLL reference clock. The IREFST bit does not update immediately after a write to the IREFS bit due to internal synchronization between clock domains. 0 Source of FLL reference clock is the external reference clock. 1 Source of FLL reference clock is the internal reference clock.

Table continues on the next page...

MCG_S field descriptions (continued)

Field	Description
3–2 CLKST	Clock Mode Status These bits indicate the current clock mode. The CLKST bits do not update immediately after a write to the CLKS bits due to internal synchronization between clock domains. 00 Encoding 0 — Output of the FLL is selected (reset default). 01 Encoding 1 — Internal reference clock is selected. 10 Encoding 2 — External reference clock is selected. 11 Encoding 3 — Output of the PLL is selected.
1 OSCINIT0	OSC Initialization This bit, which resets to 0, is set to 1 after the initialization cycles of the crystal oscillator clock have completed. After being set, the bit is cleared to 0 if the OSC is subsequently disabled. See the OSC module's detailed description for more information.
0 IRCST	Internal Reference Clock Status The IRCST bit indicates the current source for the internal reference clock select clock (IRCSCLK). The IRCST bit does not update immediately after a write to the IRCS bit due to internal synchronization between clock domains. The IRCST bit will only be updated if the internal reference clock is enabled, either by the MCG being in a mode that uses the IRC or by setting the C1[IRCLKEN] bit . 0 Source of internal reference clock is the slow clock (32 kHz IRC). 1 Source of internal reference clock is the fast clock (4 MHz IRC).

20.3.8 MCG Status and Control Register (MCG_SC)

Address: 4006_4000h base + 8h offset = 4006_4008h

Bit	7	6	5	4	3	2	1	0
Read	ATME	ATMS	ATMF	FLTPRSRV	FCRDIV			LOCS0
Write								
Reset	0	0	0	0	0	0	1	0

MCG_SC field descriptions

Field	Description
7 ATME	Automatic Trim Machine Enable Enables the Auto Trim Machine to start automatically trimming the selected Internal Reference Clock. **NOTE:** ATME deasserts after the Auto Trim Machine has completed trimming all trim bits of the IRCS clock selected by the ATMS bit. Writing to C1, C3, C4, and SC registers or entering Stop mode aborts the auto trim operation and clears this bit. 0 Auto Trim Machine disabled. 1 Auto Trim Machine enabled.

Table continues on the next page...

MCG_SC field descriptions (continued)

Field	Description
6 ATMS	Automatic Trim Machine Select Selects the IRCS clock for Auto Trim Test. 0 32 kHz Internal Reference Clock selected. 1 4 MHz Internal Reference Clock selected.
5 ATMF	Automatic Trim Machine Fail Flag Fail flag for the Automatic Trim Machine (ATM). This bit asserts when the Automatic Trim Machine is enabled, ATME=1, and a write to the C1, C3, C4, and SC registers is detected or the MCG enters into any Stop mode. A write to ATMF clears the flag. 0 Automatic Trim Machine completed normally. 1 Automatic Trim Machine failed.
4 FLTPRSRV	FLL Filter Preserve Enable This bit will prevent the FLL filter values from resetting allowing the FLL output frequency to remain the same during clock mode changes where the FLL/DCO output is still valid. (Note: This requires that the FLL reference frequency to remain the same as what it was prior to the new clock mode switch. Otherwise FLL filter and frequency values will change.) 0 FLL filter and FLL frequency will reset on changes to currect clock mode. 1 Fll filter and FLL frequency retain their previous values during new clock mode change.
3–1 FCRDIV	Fast Clock Internal Reference Divider Selects the amount to divide down the fast internal reference clock. The resulting frequency will be in the range 31.25 kHz to 4 MHz (Note: Changing the divider when the Fast IRC is enabled is not supported). 000 Divide Factor is 1 001 Divide Factor is 2. 010 Divide Factor is 4. 011 Divide Factor is 8. 100 Divide Factor is 16 101 Divide Factor is 32 110 Divide Factor is 64 111 Divide Factor is 128.
0 LOCS0	OSC0 Loss of Clock Status The LOCS0 indicates when a loss of OSC0 reference clock has occurred. The LOCS0 bit only has an effect when CME0 is set. This bit is cleared by writing a logic 1 to it when set. 0 Loss of OSC0 has not occurred. 1 Loss of OSC0 has occurred.

20.3.9 MCG Auto Trim Compare Value High Register (MCG_ATCVH)

Address: 4006_4000h base + Ah offset = 4006_400Ah

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

MCG_ATCVH field descriptions

Field	Description
7–0 ATCVH	ATM Compare Value High Values are used by Auto Trim Machine to compare and adjust Internal Reference trim values during ATM SAR conversion.

20.3.10 MCG Auto Trim Compare Value Low Register (MCG_ATCVL)

Address: 4006_4000h base + Bh offset = 4006_400Bh

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

MCG_ATCVL field descriptions

Field	Description
7–0 ATCVL	ATM Compare Value Low Values are used by Auto Trim Machine to compare and adjust Internal Reference trim values during ATM SAR conversion.

20.3.11 MCG Control 7 Register (MCG_C7)

Address: 4006_4000h base + Ch offset = 4006_400Ch

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

MCG_C7 field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–2 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
0 OSCSEL	MCG OSC Clock Select Selects the MCG FLL external reference clock 0 Selects Oscillator (OSCCLK). 1 Selects 32 kHz RTC Oscillator.

20.3.12 MCG Control 8 Register (MCG_C8)

Address: 4006_4000h base + Dh offset = 4006_400Dh

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

MCG_C8 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 LOLRE	PLL Loss of Lock Reset Enable Determines if an interrupt or a reset request is made following a PLL loss of lock. 0 Interrupt request is generated on a PLL loss of lock indication. The PLL loss of lock interrupt enable bit must also be set to generate the interrupt request. 1 Generate a reset request on a PLL loss of lock indication.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–1 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.

20.3.13 MCG Control 10 Register (MCG_C10)

Address: 4006_4000h base + Fh offset = 4006_400Fh

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

MCG_C10 field descriptions

Field	Description
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

20.4 Functional description

20.4.1 MCG mode state diagram

The nine states of the MCG are shown in the following figure and are described in Table 20-17. The arrows indicate the permitted MCG mode transitions.

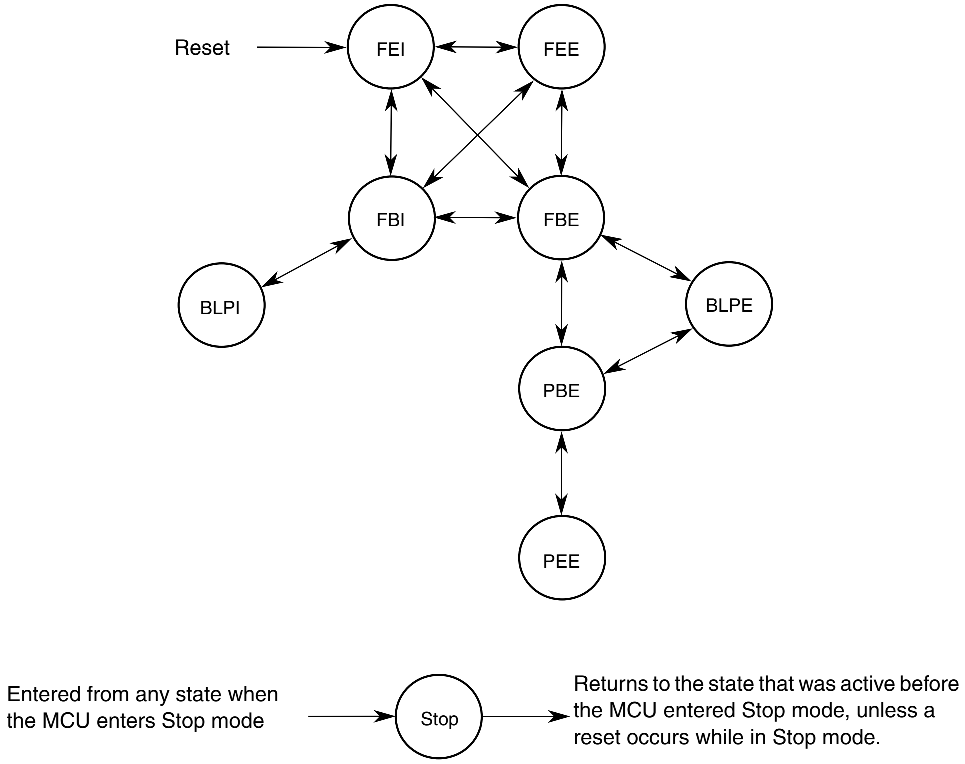


Figure 20-15. MCG mode state diagram

NOTE

- During exits from LLS or VLPS when the MCG is in PEE mode, the MCG will reset to PBE clock mode and the C1[CLKS] and S[CLKST] will automatically be set to 2'b10.
- If entering Normal Stop mode when the MCG is in PEE mode with PLLSTEN=0, the MCG will reset to PBE clock mode and C1[CLKS] and S[CLKST] will automatically be set to 2'b10.

20.4.1.1 MCG modes of operation

The MCG operates in one of the following modes.

Note

The MCG restricts transitions between modes. For the permitted transitions, see Figure 20-15.

Table 20-17. MCG modes of operation

Mode	Description
FLL Engaged Internal (FEI)	FLL engaged internal (FEI) is the default mode of operation and is entered when all the following condtions occur: • 00 is written to C1[CLKS]. • 1 is written to C1[IREFS]. • 0 is written to C6[PLLS]. In FEI mode, MCGOUTCLK is derived from the FLL clock (DCOCLK) that is controlled by the 32 kHz Internal Reference Clock (IRC). The FLL loop will lock the DCO frequency to the FLL factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the internal reference frequency. See the C4[DMX32] bit description for more details. In FEI mode, the PLL is disabled in a low-power state unless C5[PLLCLKEN] is set .
FLL Engaged External (FEE)	FLL engaged external (FEE) mode is entered when all the following conditions occur: • 00 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • C1[FRDIV] must be written to divide external reference clock to be within the range of 31.25 kHz to 39.0625 kHz • 0 is written to C6[PLLS]. In FEE mode, MCGOUTCLK is derived from the FLL clock (DCOCLK) that is controlled by the external reference clock. The FLL loop will lock the DCO frequency to the FLL factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the external reference frequency, as specified by C1[FRDIV] and C2[RANGE0]. See the C4[DMX32] bit description for more details. In FEE mode, the PLL is disabled in a low-power state unless C5[PLLCLKEN] is set .
FLL Bypassed Internal (FBI)	FLL bypassed internal (FBI) mode is entered when all the following conditions occur: • 01 is written to C1[CLKS]. • 1 is written to C1[IREFS]. • 0 is written to C6[PLLS] • 0 is written to C2[LP]. In FBI mode, the MCGOUTCLK is derived either from the slow (32 kHz IRC) or fast (4 MHz IRC) internal reference clock, as selected by the C2[IRCS] bit. The FLL is operational but its output is not used. This mode is useful to allow the FLL to acquire its target frequency while the MCGOUTCLK is driven from the C2[IRCS] selected internal reference clock. The FLL clock (DCOCLK) is controlled by the slow internal reference clock, and the DCO clock frequency locks to a multiplication factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the internal reference frequency. See the C4[DMX32] bit description for more details. In FBI mode, the PLL is disabled in a low-power state unless C5[PLLCLKEN] is set .

Table continues on the next page...

Table 20-17. MCG modes of operation (continued)

Mode	Description
FLL Bypassed External (FBE)	FLL bypassed external (FBE) mode is entered when all the following conditions occur: • 10 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • C1[FRDIV] must be written to divide external reference clock to be within the range of 31.25 kHz to 39.0625 kHz. • 0 is written to C6[PLLS]. • 0 is written to C2[LP]. In FBE mode, the MCGOUTCLK is derived from the OSCSEL external reference clock. The FLL is operational but its output is not used. This mode is useful to allow the FLL to acquire its target frequency while the MCGOUTCLK is driven from the external reference clock. The FLL clock (DCOCLK) is controlled by the external reference clock, and the DCO clock frequency locks to a multiplication factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the divided external reference frequency. See the C4[DMX32] bit description for more details. In FBI mode, the PLL is disabled in a low-power state unless C5[PLLCLKEN] is set .
PLL Engaged External (PEE)	PLL Engaged External (PEE) mode is entered when all the following conditions occur: • 00 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • 1 is written to C6[PLLS]. In PEE mode, the MCGOUTCLK is derived from the output of PLL which is controlled by a external reference clock. The PLL clock frequency locks to a multiplication factor, as specified by its corresponding VDIV, times the selected PLL reference frequency, as specified by its corresponding PRDIV. The PLL's programmable reference divider must be configured to produce a valid PLL reference clock. The FLL is disabled in a low-power state.
PLL Bypassed External (PBE)	PLL Bypassed External (PBE) mode is entered when all the following conditions occur: • 10 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • 1 is written to C6[PLLS]. • 0 is written to C2[LP]. In PBE mode, MCGOUTCLK is derived from the OSCSEL external reference clock; the PLL is operational, but its output clock is not used. This mode is useful to allow the PLL to acquire its target frequency while MCGOUTCLK is driven from the external reference clock. The PLL clock frequency locks to a multiplication factor, as specified by its [VDIV], times the PLL reference frequency, as specified by its [PRDIV]. In preparation for transition to PEE, the PLL's programmable reference divider must be configured to produce a valid PLL reference clock. The FLL is disabled in a low-power state.

Table continues on the next page...

Table 20-17. MCG modes of operation (continued)

Mode	Description
Bypassed Low Power Internal (BLPI)[1]	Bypassed Low Power Internal (BLPI) mode is entered when all the following conditions occur: • 01 is written to C1[CLKS]. • 1 is written to C1[IREFS]. • 0 is written to C6[PLLS]. • 1 is written to C2[LP]. In BLPI mode, MCGOUTCLK is derived from the internal reference clock. The FLL is disabled and PLL is disabled even if C5[PLLCLKEN] is set to 1.
Bypassed Low Power External (BLPE)	Bypassed Low Power External (BLPE) mode is entered when all the following conditions occur: • 10 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • 1 is written to C2[LP]. In BLPE mode, MCGOUTCLK is derived from the OSCSEL external reference clock. The FLL is disabled and PLL is disabled even if the C5[PLLCLKEN] is set to 1.
Stop	Entered whenever the MCU enters a Stop state. The power modes are chip specific. For power mode assignments, see the chapter that describes how modules are configured and MCG behavior during Stop recovery. Entering Stop mode, the FLL is disabled, and all MCG clock signals are static except in the following case: MCGPLLCLK is active in Normal Stop mode when PLLSTEN=1 MCGPLL1CLK is active in Normal Stop mode when PLLSTEN1=1 MCGIRCLK is active in Normal Stop mode when all the following conditions become true: • C1[IRCLKEN] = 1 • C1[IREFSTEN] = 1 **NOTE:** • In VLPS Stop Mode, the MCGIRCLK can be programmed to stay enabled and continue running if C1[IRCLKEN] = 1, C1[IREFSTEN]=1, and Fast IRC clock is selected (C2[IRCS] = 1) **NOTE:** • When entering Low Power Stop modes (LLS or VLPS) from PEE mode, on exit the MCG clock mode is forced to PBE clock mode. C1[CLKS] and S[CLKST] will be configured to 2’b10if entering from PEE mode or to 2’b01 if entering from PEI mode, C5[PLLSTEN0] will be force to 1'b0 and S[LOCK] bit will be cleared without setting S[LOLS]. • When entering Normal Stop mode from PEE mode and if C5[PLLSTEN]=0, on exit the MCG clock mode is forced to PBE mode, the C1[CLKS] and S[CLKST] will be configured to 2’b10 and S[LOCK] bit will clear without setting S[LOLS]. If C5[PLLSTEN]=1, the S[LOCK] bit will not get cleared and on exit the MCG will continue to run in PEE mode.

1. If entering VLPR mode, MCG has to be configured and enter BLPE mode or BLPI mode with the Fast IRC clock selected (C2[IRCS]=1). After it enters VLPR mode, writes to any of the MCG control registers that can cause an MCG clock mode switch to a non low power clock mode must be avoided.

NOTE

For the chip-specific modes of operation, see the power management chapter of this MCU.

20.4.1.2 MCG mode switching

C1[IREFS] can be changed at any time, but the actual switch to the newly selected reference clocks is shown by S[IREFST]. When switching between engaged internal and engaged external modes, the FLL will begin locking again after the switch is completed.

C1[CLKS] can also be changed at any time, but the actual switch to the newly selected clock is shown by S[CLKST]. If the newly selected clock is not available, the previous clock will remain selected.

The C4[DRST_DRS] write bits can be changed at any time except when C2[LP] bit is 1. If C4[DRST_DRS] write bits are changed while in FLL engaged internal (FEI) or FLL engaged external (FEE), the MCGOUTCLK will switch to the new selected DCO range within three clocks of the selected DCO clock. After switching to the new DCO, the FLL remains unlocked for several reference cycles. DCO startup time is equal to the FLL acquisition time. After the selected DCO startup time is over, the FLL is locked. The completion of the switch is shown by the C4[DRST_DRS] read bits.

20.4.2 Low-power bit usage

C2[LP] is provided to allow the FLL or PLL to be disabled and thus conserve power when these systems are not being used. C4[DRST_DRS] can not be written while C2[LP] is 1. However, in some applications, it may be desirable to enable the FLL or PLL and allow it to lock for maximum accuracy before switching to an engaged mode. Do this by writing 0 to C2[LP].

20.4.3 MCG Internal Reference Clocks

This module supports two internal reference clocks with nominal frequencies of 32 kHz (slow IRC) and 4 MHz (fast IRC). The fast IRC frequency can be divided down by programming of the FCRDIV to produce a frequency range of 32 kHz to 4 MHz.

20.4.3.1 MCG Internal Reference Clock

The MCG Internal Reference Clock (MCGIRCLK) provides a clock source for other on-chip peripherals and is enabled when C1[IRCLKEN]=1. When enabled, MCGIRCLK is driven by either the fast internal reference clock (4 MHz IRC which can be divided down by the FRDIV factors) or the slow internal reference clock (32 kHz IRC). The IRCS clock frequency can be re-targeted by trimming the period of its IRCS selected internal reference clock. This can be done by writing a new trim value to the C3[SCTRIM]:C4[SCFTRIM] bits when the slow IRC clock is selected or by writing a new trim value to C4[FCTRIM]:C2[FCFTRIM] when the fast IRC clock is selected. The internal reference clock period is proportional to the trim value written. C3[SCTRIM]:C4[SCFTRIM] (if C2[IRCS]=0) and C4[FCTRIM]:C2[FCFTRIM] (if C2[IRCS]=1) bits affect the MCGOUTCLK frequency if the MCG is in FBI or BLPI modes. C3[SCTRIM]:C4[SCFTRIM] (if C2[IRCS]=0) bits also affect the MCGOUTCLK frequency if the MCG is in FEI mode.

Additionally, this clock can be enabled in Stop mode by setting C1[IRCLKEN] and C1[IREFSTEN], otherwise this clock is disabled in Stop mode.

20.4.4 External Reference Clock

The MCG module can support an external reference clock in all modes. See the device datasheet for external reference frequency range. When C1[IREFS] is set, the external reference clock will not be used by the FLL or PLL. In these modes, the frequency can be equal to the maximum frequency the chip-level timing specifications will support.

If any of the CME bits are asserted the slow internal reference clock is enabled along with the enabled external clock monitor. For the case when C6[CME0]=1, a loss of clock is detected if the OSC0 external reference falls below a minimum frequency (f_{loc_high} or f_{loc_low} depending on C2[RANGE0]).

NOTE

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

On detecting a loss-of-clock event, the MCU generates a system reset if the respective LOCRE bit is set. Otherwise the MCG sets the respective LOCS bit and the MCG generates a LOCS interrupt request. In the case where a OSC loss of clock is detected, the PLL LOCK status bit is cleared.

20.4.5 MCG Fixed Frequency Clock

The MCG Fixed Frequency Clock (MCGFFCLK) provides a fixed frequency clock source for other on-chip peripherals; see the block diagram. This clock is driven by either the slow clock from the internal reference clock generator or the external reference clock from the Crystal Oscillator, divided by the FLL reference clock divider. The source of MCGFFCLK is selected by C1[IREFS].

This clock is synchronized to the peripheral bus clock and is valid only when its frequency is not more than 1/8 of the MCGOUTCLK frequency. When it is not valid, it is disabled and held high. The MCGFFCLK is not available when the MCG is in BLPI mode. This clock is also disabled in Stop mode. The FLL reference clock must be set within the valid frequency range for the MCGFFCLK.

20.4.6 MCG PLL clock

The MCG PLL Clock (MCGPLLCLK) is available depending on the device's configuration of the MCG module. For more details, see the clock distribution chapter of this MCU. The MCGPLLCLK is prevented from coming out of the MCG until it is enabled and S[LOCK0] is set.

20.4.7 MCG Auto TRIM (ATM)

The MCG Auto Trim (ATM) is a MCG feature that when enabled, it configures the MCG hardware to automatically trim the MCG Internal Reference Clocks using an external clock as a reference. The selection between which MCG IRC clock gets tested and enabled is controlled by the ATC[ATMS] control bit (ATC[ATMS]=0 selects the 32 kHz IRC and ATC[ATMS]=1 selects the 4 MHz IRC). If 4 MHz IRC is selected for the ATM, a divide by 128 is enabled to divide down the 4 MHz IRC to a range of 31.250 kHz.

When MCG ATM is enabled by writing ATC[ATME] bit to 1, The ATM machine will start auto trimming the selected IRC clock. During the autotrim process, ATC[ATME] will remain asserted and will deassert after ATM is completed or an abort occurs. The MCG ATM is aborted if a write to any of the following control registers is detected : C1, C3, C4, or ATC or if Stop mode is entered. If an abort occurs, ATC[ATMF] fail flag is asserted.

The ATM machine uses the bus clock as the external reference clock to perform the IRC auto-trim. Therefore, it is required that the MCG is configured in a clock mode where the reference clock used to generate the system clock is the external reference clock such as

FBE clock mode. The MCG must not be configured in a clock mode where selected IRC ATM clock is used to generate the system clock. The bus clock is also required to be running with in the range of 8–16 MHz.

To perform the ATM on the selected IRC, the ATM machine uses the successive approximation technique to adjust the IRC trim bits to generate the desired IRC trimmed frequency. The ATM SARs each of the ATM IRC trim bits starting with the MSB. For each trim bit test, the ATM uses a pulse that is generated by the ATM selected IRC clock to enable a counter that counts number of ATM external clocks. At end of each trim bit, the ATM external counter value is compared to the ATCV[15:0] register value. Based on the comparison result, the ATM trim bit under test will get cleared or stay asserted. This is done until all trim bits have been tested by ATM SAR machine.

Before the ATM can be enabled, the ATM expected count needs to be derived and stored into the ATCV register. The ATCV expected count is derived based on the required target Internal Reference Clock (IRC) frequency, and the frequency of the external reference clock using the following formula:

$$\text{ATCV Expected Count Value} = 21 * (Fe/Fr)$$

- Fr = Target Internal Reference Clock (IRC) Trimmed Frequency
- Fe = External Clock Frequency

If the auto trim is being performed on the 4 MHz IRC, the calculated expected count value must be multiplied by 128 before storing it in the ATCV register. Therefore, the ATCV Expected Count Value for trimming the 4 MHz IRC is calculated using the following formula.

$$\text{Expected Count Value} = (Fe/Fr) * 21 * (128)$$

20.5 Initialization / Application information

This section describes how to initialize and configure the MCG module in an application.

The following sections include examples on how to initialize the MCG and properly switch between the various available modes.

20.5.1 MCG module initialization sequence

The MCG comes out of reset configured for FEI mode.

The internal reference will stabilize in $t_{irefsts}$ microseconds before the FLL can acquire lock. As soon as the internal reference is stable, the FLL will acquire lock in $t_{fll\_acquire}$ milliseconds.

20.5.1.1 Initializing the MCG

Because the MCG comes out of reset in FEI mode, the only MCG modes that can be directly switched to upon reset are FEE, FBE, and FBI modes (see Figure 20-15). Reaching any of the other modes requires first configuring the MCG for one of these three intermediate modes. Care must be taken to check relevant status bits in the MCG status register reflecting all configuration changes within each mode.

To change from FEI mode to FEE or FBE modes, follow this procedure:

1. Enable the external clock source by setting the appropriate bits in C2 register.
2. Write to C1 register to select the clock mode.
    - If entering FEE mode, set C1[FRDIV] appropriately, clear C1[IREFS] bit to switch to the external reference, and leave C1[CLKS] at 2'b00 so that the output of the FLL is selected as the system clock source.
    - If entering FBE, clear C1[IREFS] to switch to the external reference and change C1[CLKS] to 2'b10 so that the external reference clock is selected as the system clock source. The C1[FRDIV] bits should also be set appropriately here according to the external reference frequency to keep the FLL reference clock in the range of 31.25 kHz to 39.0625 kHz. Although the FLL is bypassed, it is still on in FBE mode.
    - The internal reference can optionally be kept running by setting C1[IRCLKEN]. This is useful if the application will switch back and forth between internal and external modes. For minimum power consumption, leave the internal reference disabled while in an external clock mode.
3. Once the proper configuration bits have been set, wait for the affected bits in the MCG status register to be changed appropriately, reflecting that the MCG has moved into the proper mode.
    - If the MCG is in FEE, FBE, PEE, PBE, or BLPE mode, and C2[EREFS0] was also set in step 1, wait here for S[OSCINIT0] bit to become set indicating that the external clock source has finished its initialization cycles and stabilized.

- If in FEE mode, check to make sure S[IREFST] is cleared before moving on.
- If in FBE mode, check to make sure S[IREFST] is cleared and S[CLKST] bits have changed to 2'b10 indicating the external reference clock has been appropriately selected. Although the FLL is bypassed, it is still on in FBE mode.

4. Write to the C4 register to determine the DCO output (MCGFLLCLK) frequency range.
   - By default, with C4[DMX32] cleared to 0, the FLL multiplier for the DCO output is 640. For greater flexibility, if a mid-low-range FLL multiplier of 1280 is desired instead, set C4[DRST_DRS] bits to 2'b01 for a DCO output frequency of 40 MHz. If a mid high-range FLL multiplier of 1920 is desired instead, set the C4[DRST_DRS] bits to 2'b10 for a DCO output frequency of 60 MHz. If a high-range FLL multiplier of 2560 is desired instead, set the C4[DRST_DRS] bits to 2'b11 for a DCO output frequency of 80 MHz.
   - When using a 32.768 kHz external reference, if the maximum low-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b00 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 732 will be 24 MHz.
   - When using a 32.768 kHz external reference, if the maximum mid-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b01 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 1464 will be 48 MHz.
   - When using a 32.768 kHz external reference, if the maximum mid high-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b10 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 2197 will be 72 MHz.
   - When using a 32.768 kHz external reference, if the maximum high-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b11 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 2929 will be 96 MHz.
5. Wait for the FLL lock time to guarantee FLL is running at new C4[DRST_DRS] and C4[DMX32] programmed frequency.

To change from FEI clock mode to FBI clock mode, follow this procedure:

1. Change C1[CLKS] bits in C1 register to 2'b01 so that the internal reference clock is selected as the system clock source.
2. Wait for S[CLKST] bits in the MCG status register to change to 2'b01, indicating that the internal reference clock has been appropriately selected.
3. Write to the C2 register to determine the IRCS output (IRCSCLK) frequency range.
   - By default, with C2[IRCS] cleared to 0, the IRCS selected output clock is the slow internal reference clock (32 kHz IRC). If the faster IRC is desired, set C2[IRCS] to 1 for a IRCS clock derived from the 4 MHz IRC source.

20.5.2 Using a 32.768 kHz reference

In FEE and FBE modes, if using a 32.768 kHz external reference, at the default FLL multiplication factor of 640, the DCO output (MCGFLLCLK) frequency is 20.97 MHz at low-range.

If C4[DRST_DRS] bits are set to 2'b01, the multiplication factor is doubled to 1280, and the resulting DCO output frequency is 41.94 MHz at mid-low-range. If C4[DRST_DRS] bits are set to 2'b10, the multiplication factor is set to 1920, and the resulting DCO output frequency is 62.91 MHz at mid high-range. If C4[DRST_DRS] bits are set to 2'b11, the multiplication factor is set to 2560, and the resulting DCO output frequency is 83.89 MHz at high-range.

In FBI and FEI modes, setting C4[DMX32] bit is not recommended. If the internal reference is trimmed to a frequency above 32.768 kHz, the greater FLL multiplication factor could potentially push the microcontroller system clock out of specification and damage the part.

20.5.3 MCG mode switching

When switching between operational modes of the MCG, certain configuration bits must be changed in order to properly move from one mode to another.

Each time any of these bits are changed (C6[PLLS], C1[IREFS], C1[CLKS], C2[IRCS], or C2[EREFS0]), the corresponding bits in the MCG status register (PLLST, IREFST, CLKST, IRCST, or OSCINIT) must be checked before moving on in the application software.

Additionally, care must be taken to ensure that the reference clock divider (C1[FRDIV] and C5[PRDIV0]) is set properly for the mode being switched to. For instance, in PEE mode, if using a 4 MHz crystal, C5[PRDIV0] must be set to 5'b000 (divide-by-1) or 5'b001 (divide-by-2) to divide the external reference down to the required frequency between 2 and 4 MHz.

In FBE, FEE, FBI, and FEI modes, at any time, the application can switch the FLL multiplication factor between 640, 1280, 1920, and 2560 with C4[DRST_DRS] bits. Writes to C4[DRST_DRS] bits will be ignored if C2[LP]=1.

The table below shows MCGOUTCLK frequency calculations using C1[FRDIV], C5[PRDIV0], and C6[VDIV0] settings for each clock mode.

Table 20-18. MCGOUTCLK Frequency Calculation Options

Clock Mode	$f_{MCGOUTCLK}$[1]	Note
FEI (FLL engaged internal)	(f_{int} * F)	Typical $f_{MCGOUTCLK}$ = 21 MHz immediately after reset.
FEE (FLL engaged external)	(f_{ext} / FLL_R) *F	f_{ext} / FLL_R must be in the range of 31.25 kHz to 39.0625 kHz
FBE (FLL bypassed external)	OSCCLK	OSCCLK / FLL_R must be in the range of 31.25 kHz to 39.0625 kHz
FBI (FLL bypassed internal)	MCGIRCLK	Selectable between slow and fast IRC
PEE (PLL engaged external)	(OSCCLK / PLL_R) * M	OSCCLK / PLL_R must be in the range of 2 – 4 MHz
PBE (PLL bypassed external)	OSCCLK	OSCCLK / PLL_R must be in the range of 2 – 4 MHz
BLPI (Bypassed low power internal)	MCGIRCLK	Selectable between slow and fast IRC
BLPE (Bypassed low power external)	OSCCLK	

1. FLL_R is the reference divider selected by the C1[FRDIV] bits, PLL_R is the reference divider selected by C5[PRDIV0] bits, F is the FLL factor selected by C4[DRST_DRS] and C4[DMX32] bits, and M is the multiplier selected by C6[VDIV0] bits.

This section will include three mode switching examples using an 4 MHz external crystal. If using an external clock source less than 2 MHz, the MCG must not be configured for any of the PLL modes (PEE and PBE).

20.5.3.1 Example 1: Moving from FEI to PEE mode: External Crystal = 4 MHz, MCGOUTCLK frequency = 48 MHz

In this example, the MCG will move through the proper operational modes from FEI to PEE to achieve 48 MHz MCGOUTCLK frequency from 4 MHz external crystal reference. First, the code sequence will be described. Then there is a flowchart that illustrates the sequence.

1. First, FEI must transition to FBE mode:
   a. C2 = 0x2C
      - C2[RANGE0] set to 2'b01 because the frequency of 4 MHz is within the high frequency range.
      - C2[HGO0] set to 1 to configure the crystal oscillator for high gain operation.
      - C2[EREFS0] set to 1, because a crystal is being used.

   b. C1 = 0x90
      - C1[CLKS] set to 2'b10 to select external reference clock as system clock source
      - C1[FRDIV] set to 3'b010, or divide-by-128 because 4 MHz / 128 = 31.25 kHz which is in the 31.25 kHz to 39.0625 kHz range required by the FLL
      - C1[IREFS] cleared to 0, selecting the external reference clock and enabling the external oscillator.

   c. Loop until S[OSCINIT0] is 1, indicating the crystal selected by C2[EREFS0] has been initialized.
   d. Loop until S[IREFST] is 0, indicating the external reference is the current source for the reference clock.
   e. Loop until S[CLKST] is 2'b10, indicating that the external reference clock is selected to feed MCGOUTCLK.
2. Then configure C5[PRDIV0] to generate correct PLL reference frequency.
   a. C5 = 0x01
      - C5[PRDIV] set to 5'b00001, or divide-by-2 resulting in a pll reference frequency of 4MHz/2 = 2 MHz.
3. Then, FBE must transition either directly to PBE mode or first through BLPE mode and then to PBE mode:

a. BLPE: If a transition through BLPE mode is desired, first set C2[LP] to 1.

b. BLPE/PBE: C6 = 0x40

- C6[PLLS] set to 1, selects the PLL. At this time, with a C1[PRDIV] value of 2'b001, the PLL reference divider is 2 (see PLL External Reference Divide Factor table), resulting in a reference frequency of 4 MHz/ 2 = 2 MHz. In BLPE mode, changing the C6[PLLS] bit only prepares the MCG for PLL usage in PBE mode.
- C6[VDIV] set to 5'b00000, or multiply-by-24 because 2 MHz reference * 24 = 48 MHz. In BLPE mode, the configuration of the VDIV bits does not matter because the PLL is disabled. Changing them only sets up the multiply value for PLL usage in PBE mode.

c. BLPE: If transitioning through BLPE mode, clear C2[LP] to 0 here to switch to PBE mode.

d. PBE: Loop until S[PLLST] is set, indicating that the current source for the PLLS clock is the PLL.

e. PBE: Then loop until S[LOCK0] is set, indicating that the PLL has acquired lock.

4. Lastly, PBE mode transitions into PEE mode:

a. C1 = 0x10

- C1[CLKS] set to 2'b00 to select the output of the PLL as the system clock source.

b. Loop until S[CLKST] are 2'b11, indicating that the PLL output is selected to feed MCGOUTCLK in the current clock mode.

- Now, with PRDIV of divide-by-2, and C6[VDIV] of multiply-by-24, MCGOUTCLK = [(4 MHz / 2) * 24] = 48 MHz.

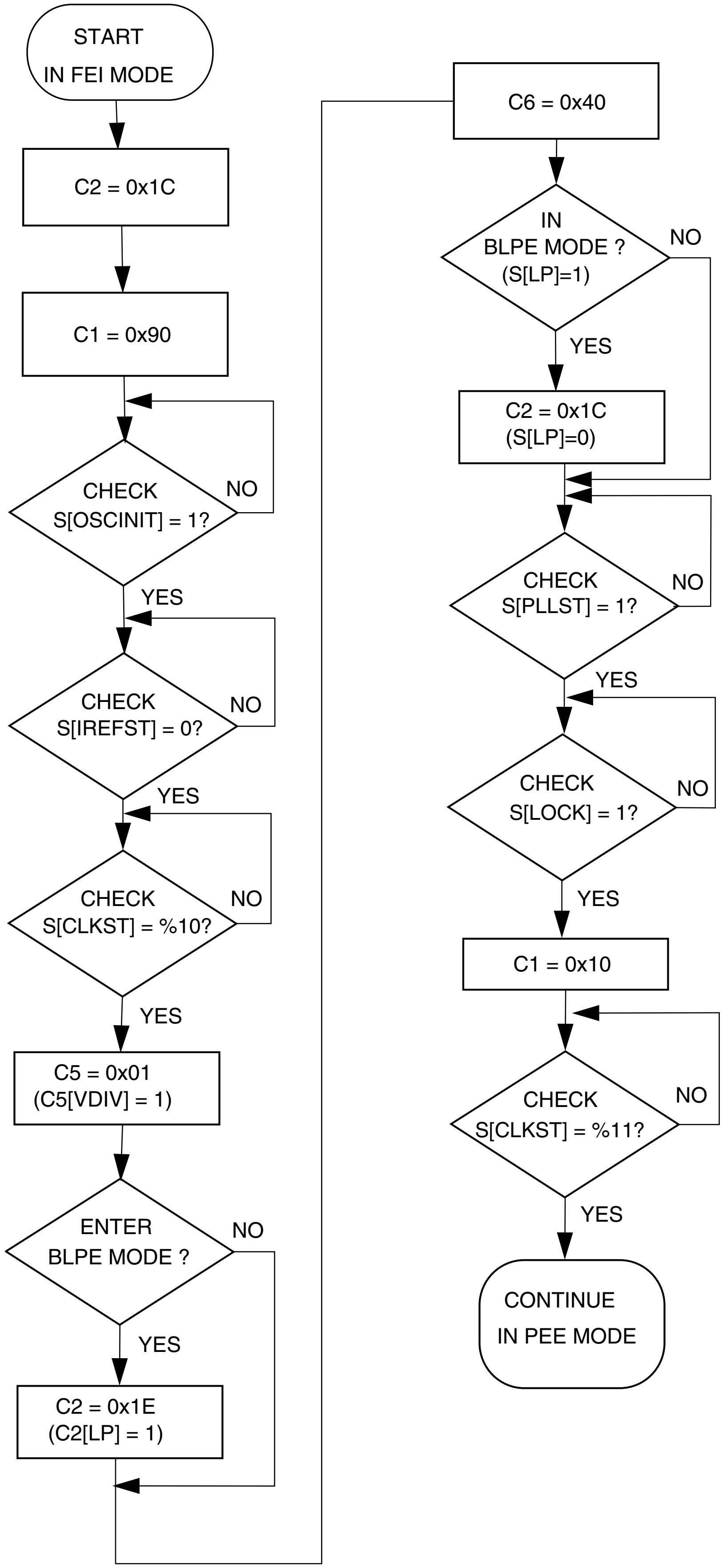


Figure 20-16. Flowchart of FEI to PEE mode transition using an 4 MHz crystal

20.5.3.2 Example 2: Moving from PEE to BLPI mode: MCGOUTCLK frequency =32 kHz

In this example, the MCG will move through the proper operational modes from PEE mode with a 4 MHz crystal configured for a 48 MHz MCGOUTCLK frequency (see previous example) to BLPI mode with a 32 kHz MCGOUTCLK frequency. First, the code sequence will be described. Then there is a flowchart that illustrates the sequence.

1. First, PEE must transition to PBE mode:
   a. C1 = 0x90
      - C1[CLKS] set to 2'b10 to switch the system clock source to the external reference clock.

   b. Loop until S[CLKST] are 2'b10, indicating that the external reference clock is selected to feed MCGOUTCLK.
2. Then, PBE must transition either directly to FBE mode or first through BLPE mode and then to FBE mode:
   a. BLPE: If a transition through BLPE mode is desired, first set C2[LP] to 1.

   b. BLPE/FBE: C6 = 0x00
      - C6[PLLS] clear to 0 to select the FLL. At this time, with C1[FRDIV] value of 3'b010, the FLL divider is set to 128, resulting in a reference frequency of 4 MHz / 128 = 31.25 kHz. If C1[FRDIV] was not previously set to 3'b010 (necessary to achieve required 31.25–39.06 kHz FLL reference frequency with an 4 MHz external source frequency), it must be changed prior to clearing C6[PLLS] bit. In BLPE mode,changing this bit only prepares the MCG for FLL usage in FBE mode. With C6[PLLS] = 0, the C6[VDIV] value does not matter.

   c. BLPE: If transitioning through BLPE mode, clear C2[LP] to 0 here to switch to FBE mode.

   d. FBE: Loop until S[PLLST] is cleared, indicating that the current source for the PLLS clock is the FLL.
3. Next, FBE mode transitions into FBI mode:
   a. C1 = 0x54
      - C1[CLKS] set to 2'b01 to switch the system clock to the internal reference clock.

- C1[IREFS] set to 1 to select the internal reference clock as the reference clock source.
- C1[FRDIV] remain unchanged because the reference divider does not affect the internal reference.

b. Loop until S[IREFST] is 1, indicating the internal reference clock has been selected as the reference clock source.

c. Loop until S[CLKST] are 2'b01, indicating that the internal reference clock is selected to feed MCGOUTCLK.

4. Lastly, FBI transitions into BLPI mode.

a. C2 = 0x02

- C2[LP] is 1
- C2[RANGE0], C2[HGO0], C2[EREFS0], C1[IRCLKEN], and C1[IREFSTEN] bits are ignored when the C1[IREFS] bit is set. They can remain set, or be cleared at this point.

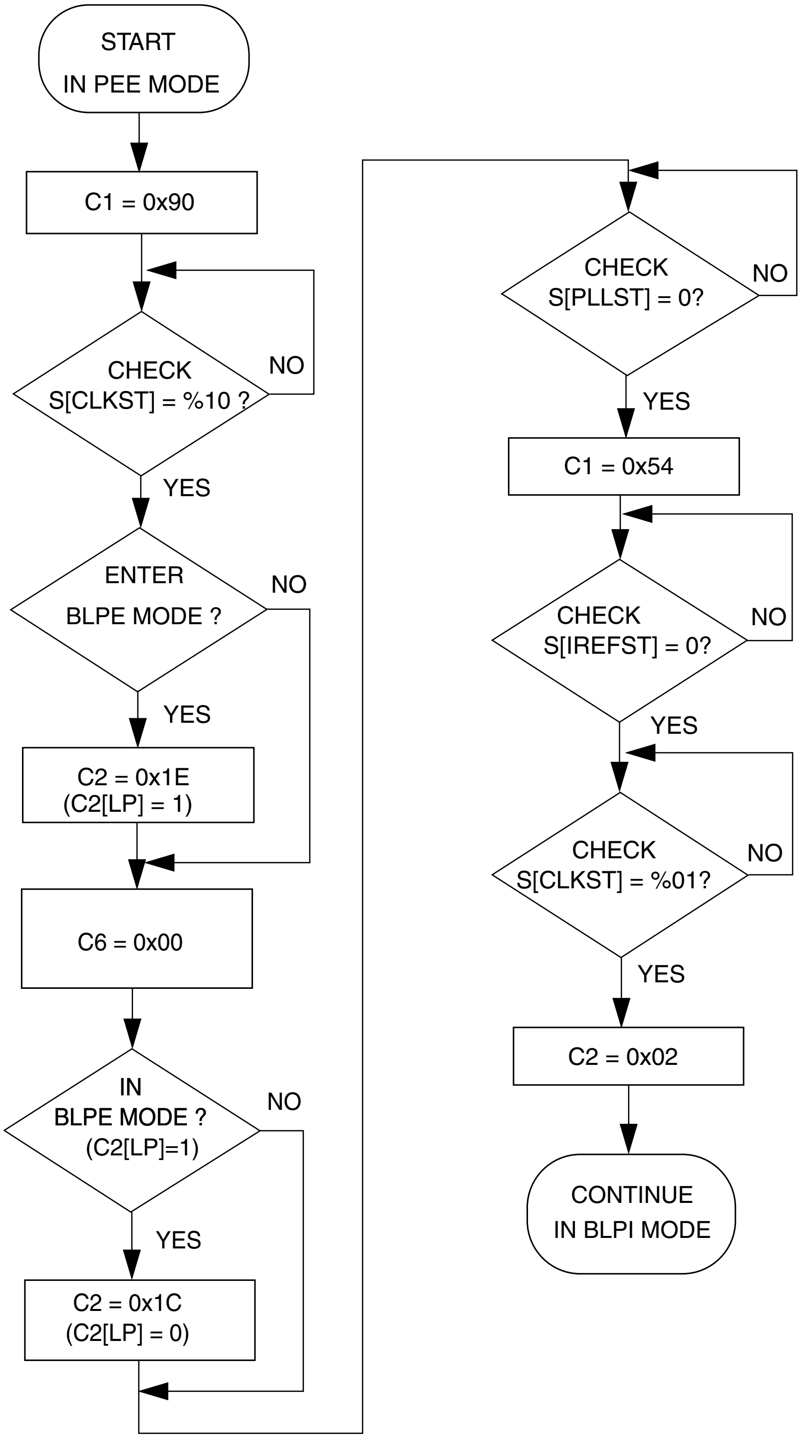


Figure 20-17. Flowchart of PEE to BLPI mode transition using an 4 MHz crystal

Chapter 21
Oscillator (OSC)

21.1 Introduction

NOTE

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

The OSC module is a crystal oscillator. The module, in conjunction with an external crystal or resonator, generates a reference clock for the MCU.

21.2 Features and Modes

Key features of the module are listed here.

- Supports 32 kHz crystals (Low Range mode)
- Supports 3–8 MHz, 8–32 MHz crystals and resonators (High Range mode)
- Automatic Gain Control (AGC) to optimize power consumption in high frequency ranges 3–8 MHz, 8–32 MHz using low-power mode
- High gain option in frequency ranges: 32 kHz, 3–8 MHz, and 8–32 MHz
- Voltage and frequency filtering to guarantee clock frequency and stability
- Optionally external input bypass clock from EXTAL signal directly
- One clock for MCU clock system
- Two clocks for on-chip peripherals that can work in Stop modes

Functional Description describes the module's operation in more detail.

21.3 Block Diagram

The OSC module uses a crystal or resonator to generate three filtered oscillator clock signals.Three clocks are output from OSC module: OSCCLK for MCU system, OSCERCLK for on-chip peripherals, and OSC32KCLK. The OSCCLK can only work in run mode. OSCERCLK and OSC32KCLK can work in low power modes. For the clock source assignments, refer to the clock distribution information of this MCU.

Refer to the chip configuration details for the external reference clock source in this MCU.

The figure found here shows the block diagram of the OSC module.

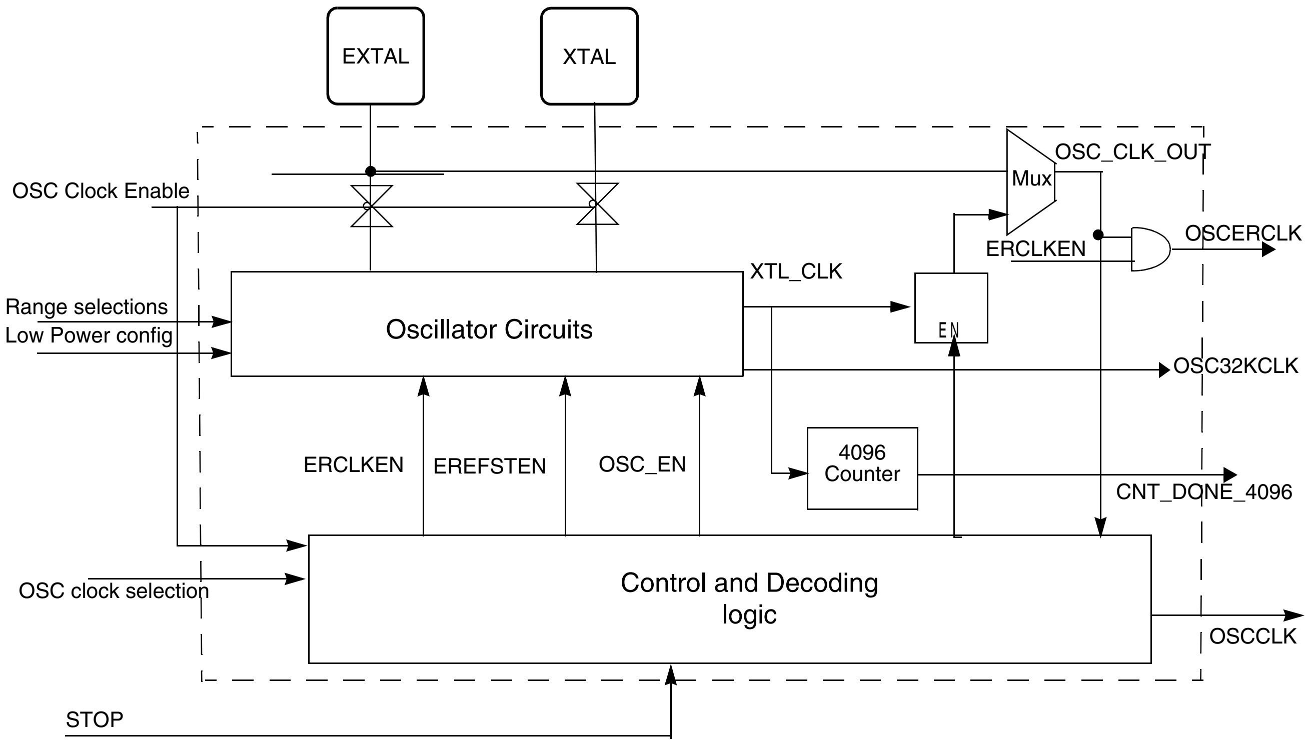


Figure 21-1. OSC Module Block Diagram

21.4 OSC Signal Descriptions

The table found here shows the user-accessible signals available for the OSC module.

Refer to signal multiplexing information for this MCU for more details.

Table 21-1. OSC Signal Descriptions

Signal	Description	I/O
EXTAL	External clock/Oscillator input	I
XTAL	Oscillator output	O

21.5 External Crystal / Resonator Connections

The connections for a crystal/resonator frequency reference are shown in the figures found here.

When using low-frequency, low-power mode, the only external component is the crystal or ceramic resonator itself. In the other oscillator modes, load capacitors (C_x, C_y) and feedback resistor (R_F) are required. The following table shows all possible connections.

Table 21-2. External Caystal/Resonator Connections

Oscillator Mode	Connections
Low-frequency (32 kHz), low-power	Connection 1[1]
Low-frequency (32 kHz), high-gain	Connection 2/Connection 3[2]
High-frequency (3~32 MHz), low-power	Connection 3[1]
High-frequency (3~32 MHz), high-gain	Connection 3

1. With the low-power mode, the oscillator has the internal feedback resistor R_F. Therefore, the feedback resistor must not be externally with the Connection 3.
2. When the load capacitors (Cx, Cy) are greater than 30 pF, use Connection 3.

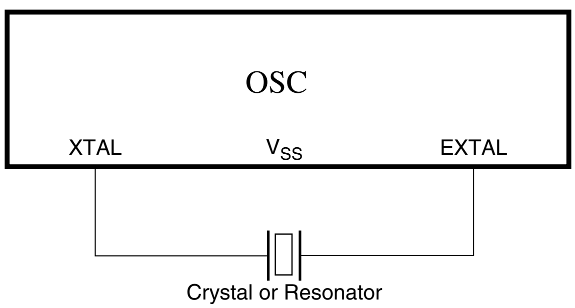


Figure 21-2. Crystal/Ceramic Resonator Connections - Connection 1

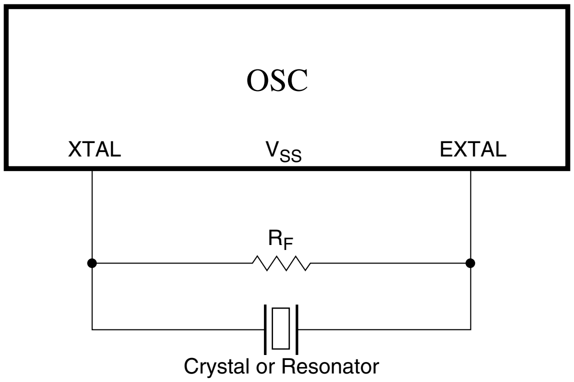


Figure 21-3. Crystal/Ceramic Resonator Connections - Connection 2

NOTE

Connection 1 and Connection 2 should use internal capacitors as the load of the oscillator by configuring the CR[SCxP] bits.

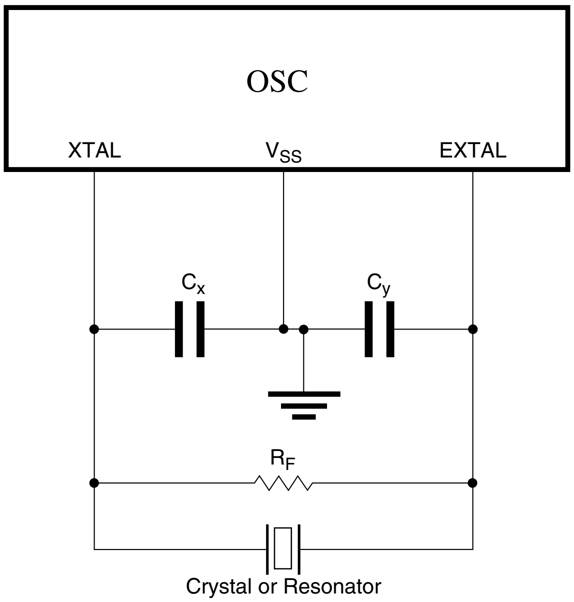


Figure 21-4. Crystal/Ceramic Resonator Connections - Connection 3

21.6 External Clock Connections

In external clock mode, the pins can be connected as shown in the figure found here.

NOTE

XTAL can be used as a GPIO when the GPIO alternate function is configured for it.

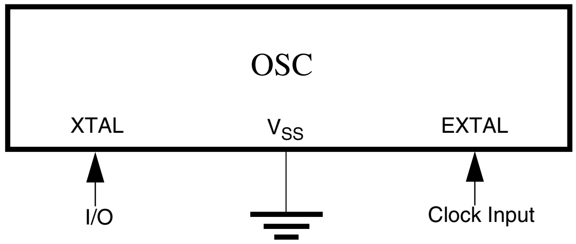


Figure 21-5. External Clock Connections

21.7 Memory Map/Register Definitions

Some oscillator module register bits are typically incorporated into other peripherals such as MCG or SIM.

21.7.1 OSC Memory Map/Register Definition

OSC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_5000	OSC Control Register (OSC0_CR)	8	R/W	00h	21.71.1/369

21.71.1 OSC Control Register (OSCx_CR)

NOTE

After OSC is enabled and starts generating the clocks, the configurations such as low power and frequency range, must not be changed.

Address: 4006_5000h base + 0h offset = 4006_5000h

Bit	7	6	5	4	3	2	1	0
Read	ERCLKEN	0	EREFSTEN	0	SC2P	SC4P	SC8P	SC16P
Write								
Reset	0	0	0	0	0	0	0	0

OSCx_CR field descriptions

Field	Description
7 ERCLKEN	External Reference Enable Enables external reference clock (OSCERCLK).

Table continues on the next page...

OSCx_CR field descriptions (continued)

Field	Description
	0 External reference clock is inactive. 1 External reference clock is enabled.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 EREFSTEN	External Reference Stop Enable Controls whether or not the external reference clock (OSCERCLK) remains enabled when MCU enters Stop mode. 0 External reference clock is disabled in Stop mode. 1 External reference clock stays enabled in Stop mode if ERCLKEN is set before entering Stop mode.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 SC2P	Oscillator 2 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 2 pF capacitor to the oscillator load.
2 SC4P	Oscillator 4 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 4 pF capacitor to the oscillator load.
1 SC8P	Oscillator 8 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 8 pF capacitor to the oscillator load.
0 SC16P	Oscillator 16 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 16 pF capacitor to the oscillator load.

21.8 Functional Description

Functional details of the module can be found here.

21.8.1 OSC module states

The states of the OSC module are shown in the following figure. The states and their transitions between each other are described in this section.

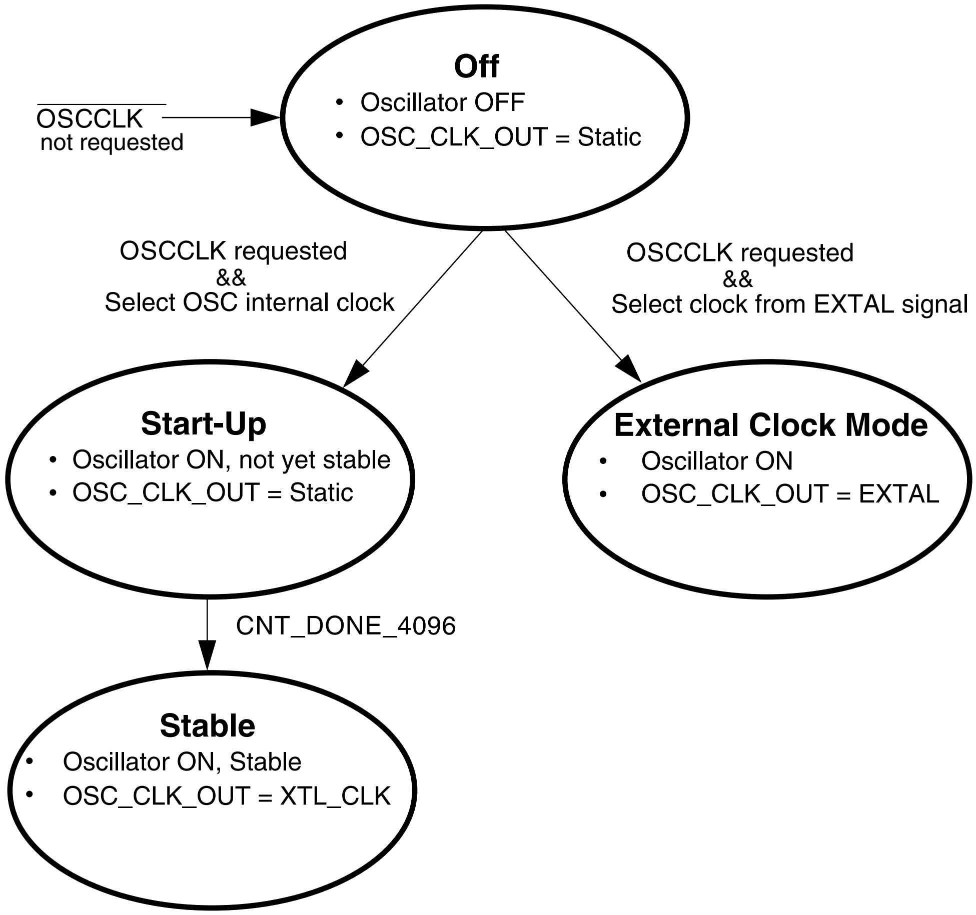


Figure 21-8. OSC Module state diagram

NOTE

XTL_CLK is the clock generated internally from OSC circuits.

21.8.1.1 Off

The OSC enters the Off state when the system does not require OSC clocks. Upon entering this state, XTL_CLK is static unless OSC is configured to select the clock from the EXTAL pad by clearing the external reference clock selection bit. For details regarding the external reference clock source in this MCU, refer to the chip configuration details. The EXTAL and XTAL pins are also decoupled from all other oscillator circuitry in this state. The OSC module circuitry is configured to draw minimal current.

21.8.1.2 Oscillator startup

The OSC enters startup state when it is configured to generate clocks (internally the OSC_EN transitions high) using the internal oscillator circuits by setting the external reference clock selection bit. In this state, the OSC module is enabled and oscillations are starting up, but have not yet stabilized. When the oscillation amplitude becomes large enough to pass through the input buffer, XTL_CLK begins clocking the counter. When the counter reaches 4096 cycles of XTL_CLK, the oscillator is considered stable and XTL_CLK is passed to the output clock OSC_CLK_OUT.

21.8.1.3 Oscillator Stable

The OSC enters stable state when it is configured to generate clocks (internally the OSC_EN transitions high) using the internal oscillator circuits by setting the external reference clock selection bit and the counter reaches 4096 cycles of XTL_CLK (when CNT_DONE_4096 is high). In this state, the OSC module is producing a stable output clock on OSC_CLK_OUT. Its frequency is determined by the external components being used.

21.8.1.4 External Clock mode

The OSC enters external clock state when it is enabled and external reference clock selection bit is cleared. For details regarding external reference clock source in this MCU, see the chip configuration details. In this state, the OSC module is set to buffer (with hysteresis) a clock from EXTAL onto the OSC_CLK_OUT. Its frequency is determined by the external clock being supplied.

21.8.2 OSC module modes

The OSC is a pierce-type oscillator that supports external crystals or resonators operating over the frequency ranges shown in Table 21-7. These modes assume the following conditions: OSC is enabled to generate clocks (OSC_EN=1), configured to generate clocks internally (MCG_C2[EREFS] = 1), and some or one of the other peripherals (MCG, Timer, and so on) is configured to use the oscillator output clock (OSC_CLK_OUT).

Table 21-7. Oscillator modes

Mode	Frequency Range
Low-frequency, high-gain	f_{osc_lo} (32.768 kHz) up to f_{osc_lo} (39.0625 kHz)

Table continues on the next page...

Table 21-7. Oscillator modes (continued)

Mode	Frequency Range
High-frequency mode1, high-gain	$f_{osc\_hi\_1}$ (3 MHz) up to $f_{osc\_hi\_1}$ (8 MHz)
High-frequency mode1, low-power	
High-frequency mode2, high-gain	$f_{osc\_hi\_2}$ (8 MHz) up to $f_{osc\_hi\_2}$ (32 MHz)
High-frequency mode2, low-power	

NOTE

For information about low power modes of operation used in this chip and their alignment with some OSC modes, see the chip's Power Management details.

21.8.2.1 Low-Frequency, High-Gain Mode

In Low-frequency, high-gain mode, the oscillator uses a simple inverter-style amplifier. The gain is set to achieve rail-to-rail oscillation amplitudes.

The oscillator input buffer in this mode is single-ended. It provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels. In this mode, the internal capacitors could be used.

21.8.2.2 Low-Frequency, Low-Power Mode

In low-frequency, low-power mode, the oscillator uses a gain control loop to minimize power consumption. As the oscillation amplitude increases, the amplifier current is reduced. This continues until a desired amplitude is achieved at steady-state. This mode provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels. In this mode, the internal capacitors could be used, the internal feedback resistor is connected, and no external resistor should be used.

In this mode, the amplifier inputs, gain-control input, and input buffer input are all capacitively coupled for leakage tolerance (not sensitive to the DC level of EXTAL).

Also in this mode, all external components except for the resonator itself are integrated, which includes the load capacitors and feeback resistor that biases EXTAL.

21.8.2.3 High-Frequency, High-Gain Mode

In high-frequency, high-gain mode, the oscillator uses a simple inverter-style amplifier. The gain is set to achieve rail-to-rail oscillation amplitudes. This mode provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels.

21.8.2.4 High-Frequency, Low-Power Mode

In high-frequency, low-power mode, the oscillator uses a gain control loop to minimize power consumption. As the oscillation amplitude increases, the amplifier current is reduced. This continues until a desired amplitude is achieved at steady-state. In this mode, no external resistor should be used.

The oscillator input buffer in this mode is differential. It provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels.

21.8.3 Counter

The oscillator output clock (OSC_CLK_OUT) is gated off until the counter has detected 4096 cycles of its input clock (XTL_CLK). After 4096 cycles are completed, the counter passes XTL_CLK onto OSC_CLK_OUT. This counting timeout is used to guarantee output clock stability.

21.8.4 Reference clock pin requirements

The OSC module requires use of both the EXTAL and XTAL pins to generate an output clock in Oscillator mode, but requires only the EXTAL pin in External clock mode. The EXTAL and XTAL pins are available for I/O. For the implementation of these pins on this device, refer to the Signal Multiplexing chapter.

21.9 Reset

There is no reset state associated with the OSC module. The counter logic is reset when the OSC is not configured to generate clocks.

There are no sources of reset requests for the OSC module.

21.10 Low power modes operation

When the MCU enters Stop modes, the OSC is functional depending on CR[ERCLKEN] and CR[EREFSETN] bit settings. If both these bits are set, the OSC is in operation.

In Low Leakage Stop (LLS) modes, the OSC holds all register settings. If CR[ERCLKEN] and CR[EREFSTEN] are set before entry to Low Leakage Stop modes, the OSC is still functional in these modes. After waking up from Very Low Leakage Stop (VLLSx) modes, all OSC register bits are reset and initialization is required through software.

21.11 Interrupts

The OSC module does not generate any interrupts.

Chapter 22
Flash Memory Controller (FMC)

22.1 Introduction

The Flash Memory Controller (FMC) is a memory acceleration unit. A list of features provided by the FMC can be found here.

- an interface between bus masters and the 32-bit program flash memory.
- a buffer and a cache that can accelerate program flash memory data transfers.

22.1.1 Overview

The Flash Memory Controller manages the interface between bus masters and the 32-bit program flash memory. The FMC receives status information detailing the configuration of the flash memory and uses this information to ensure a proper interface. The FMC supports 8-bit, 16-bit, and 32-bit read operations from the program flash memory. A write operation to program flash memory results in a bus error.

In addition, the FMC provides two separate mechanisms for accelerating the interface between bus masters and program flash memory. A 32-bit speculation buffer can prefetch the next 32-bit flash memory location, and a 4-way, 4-set program flash memory cache can store previously accessed program flash memory data for quick access times.

22.1.2 Features

The features of FMC module include:

- Interface between bus masters and the 32-bit program flash memory:
  - 8-bit, 16-bit, and 32-bit read operations to nonvolatile flash memory.
- Acceleration of data transfer from the program flash memory to the device:

- 32-bit prefetch speculation buffer for program flash accesses with controls for instruction/data access
- 4-way, 4-set, 32-bit line size program flash memory cache for a total of sixteen 32-bit entries with invalidation control

22.2 Modes of operation

The FMC operates only when a bus master accesses the program flash memory.

In terms of chip power modes:

- The FMC operates only in Run and Wait modes, including VLPR and VLPW modes.
- For any power mode where the program flash memory cannot be accessed, the FMC is disabled.

22.3 External signal description

The FMC has no external (off-chip) signals.

22.4 Memory map and register descriptions

The MCM's programming model provides control and configuration of the FMC's features.

For details, see the description of the MCM's Platform Control Register (PLACR).

22.5 Functional description

The FMC is a flash acceleration unit with flexible buffers for user configuration.

Besides managing the interface between bus masters and the program flash memory, the FMC can be used to customize the program flash memory cache and buffer to provide single-cycle system clock data access times. Whenever a hit occurs for the prefetch speculation buffer or the cache (when enabled), the requested data is transferred within a single system clock.

Upon system reset, the FMC is configured as follows:

- Flash cache is enabled.
- Instruction speculation and caching are enabled.

- Data speculation is disabled.
- Data caching is enabled.

Though the default configuration provides flash acceleration, advanced users may desire to customize the FMC buffer configurations to maximize throughput for their use cases. For example, the user may adjust the controls to enable buffering per access type (data or instruction).

NOTE

When reconfiguring the FMC, do not program the control and configuration inputs to the FMC while the program flash memory is being accessed. Instead, change them with a routine executing from RAM in supervisor mode.

Chapter 23
Flash Memory Module (FTFA)

23.1 Introduction

NOTE

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

The flash memory module includes the following accessible memory regions:

- Program flash memory for vector space and code store

Flash memory is ideal for single-supply applications, permitting in-the-field erase and reprogramming operations without the need for any external high voltage power sources.

The flash memory module includes a memory controller that executes commands to modify flash memory contents. An erased bit reads '1' and a programmed bit reads '0'. The programming operation is unidirectional; it can only move bits from the '1' state (erased) to the '0' state (programmed). Only the erase operation restores bits from '0' to '1'; bits cannot be programmed from a '0' to a '1'.

CAUTION

A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash memory location is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.

The standard shipping condition for flash memory is erased with security disabled. Data loss over time may occur due to degradation of the erased ('1') states and/or programmed ('0') states. Therefore, it is recommended that each flash block or sector be re-erased immediately prior to factory programming to ensure that the full data retention capability is achieved.

23.1.1 Features

The flash memory module includes the following features.

NOTE

See the device's Chip Configuration details for the exact amount of flash memory available on your device.

23.1.1.1 Program Flash Memory Features

- Sector size of 1 KB
- Program flash protection scheme prevents accidental program or erase of stored data
- Automated, built-in, program and erase algorithms with verify

23.1.1.2 Other Flash Memory Module Features

- Internal high-voltage supply generator for flash memory program and erase operations
- Optional interrupt generation upon flash command completion
- Supports MCU security mechanisms which prevent unauthorized access to the flash memory contents

23.1.2 Block Diagram

The block diagram of the flash memory module is shown in the following figure.

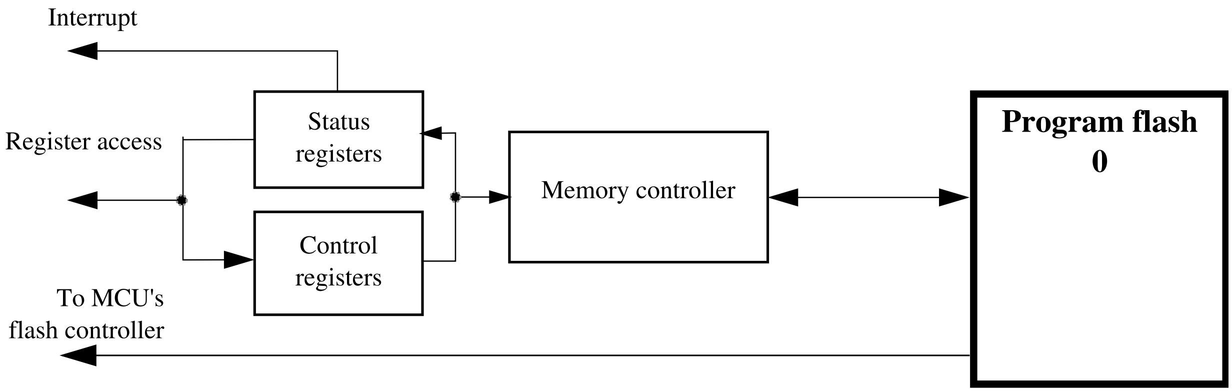


Figure 23-1. Flash Block Diagram

23.1.3 Glossary

Command write sequence — A series of MCU writes to the flash FCCOB register group that initiates and controls the execution of flash algorithms that are built into the flash memory module.

Endurance — The number of times that a flash memory location can be erased and reprogrammed.

FCCOB (Flash Common Command Object) — A group of flash registers that are used to pass command, address, data, and any associated parameters to the memory controller in the flash memory module.

Flash block — A macro within the flash memory module which provides the nonvolatile memory storage.

Flash Memory Module — All flash blocks plus a flash management unit providing high-level control and an interface to MCU buses.

IFR — Nonvolatile information register found in each flash block, separate from the main memory array.

Longword — 32 bits of data with an aligned longword having byte-address[1:0] = 00.

NVM — Nonvolatile memory. A memory technology that maintains stored data during power-off. The flash array is an NVM using NOR-type flash memory technology.

NVM Normal Mode — An NVM mode that provides basic user access to flash memory module resources. The CPU or other bus masters initiate flash program and erase operations (or other flash commands) using writes to the FCCOB register group in the flash memory module.

NVM Special Mode — An NVM mode enabling external, off-chip access to the memory resources in the flash memory 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.

Program flash — The program flash memory provides nonvolatile storage for vectors and code store.

Program flash Sector — The smallest portion of the program flash memory (consecutive addresses) that can be erased.

Retention — The length of time that data can be kept in the NVM without experiencing errors upon readout. Since erased (1) states are subject to degradation just like programmed (0) states, the data retention limit may be reached from the last erase operation (not from the programming time).

RWW— Read-While-Write. The ability to simultaneously read from one memory resource while commanded operations are active in another memory resource.

Secure — An MCU state conveyed to the flash memory module as described in the Chip Configuration details for this device. In the secure state, reading and changing NVM contents is restricted.

Word — 16 bits of data with an aligned word having byte-address[0] = 0.

23.2 External Signal Description

The flash memory module contains no signals that connect off-chip.

23.3 Memory Map and Registers

This section describes the memory map and registers for the flash memory module.

Data read from unimplemented memory space in the flash memory module is undefined. Writes to unimplemented or reserved memory space (registers) in the flash memory module are ignored.

23.3.1 Flash Configuration Field Description

The program flash memory contains a 16-byte flash configuration field that stores default protection settings (loaded on reset) and security information that allows the MCU to restrict access to the flash memory module.

Flash Configuration Field Byte Address	Size (Bytes)	Field Description
0x0_0400–0x0_0407	8	Backdoor Comparison Key. Refer to Verify Backdoor Access Key Command and Unsecuring the Chip Using Backdoor Key Access.
0x0_0408–0x0_040B	4	Program flash protection bytes. Refer to the description of the Program Flash Protection Registers (FPROT0-3).
0x0_040F	1	Reserved
0x0_040E	1	Reserved
0x0_040D	1	Flash nonvolatile option byte. Refer to the description of the Flash Option Register (FOPT).
0x0_040C	1	Flash security byte. Refer to the description of the Flash Security Register (FSEC).

23.3.2 Program Flash IFR Map

The program flash IFR is nonvolatile information memory that can be read freely, but the user has no erase and limited program capabilities (see the Read Once, Program Once, and Read Resource commands in Read Once Command, Program Once Command and Read Resource Command).

The contents of the program flash IFR are summarized in the table found here and further described in the subsequent paragraphs.

The program flash IFR is located within the program flash 0 memory block.

Address Range	Size (Bytes)	Field Description
0x00 – 0xBF	192	Reserved
0xC0 – 0xFF	64	Program Once Field

23.3.2.1 Program Once Field

The Program Once Field in the program flash IFR provides 64 bytes of user data storage separate from the program flash main array. The user can program the Program Once Field one time only as there is no program flash IFR erase mechanism available to the user. The Program Once Field can be read any number of times. This section of the program flash IFR is accessed in 4-byte records using the Read Once and Program Once commands (see Read Once Command and Program Once Command).

23.3.3 Register Descriptions

The flash memory module contains a set of memory-mapped control and status registers.

NOTE

While a command is running (FSTAT[CCIF]=0), register writes are not accepted to any register except FCNFG and FSTAT. The no-write rule is relaxed during the start-up reset sequence, prior to the initial rise of CCIF. During this initialization period the user may write any register. All register writes are also disabled (except for registers FCNFG and FSTAT) whenever an erase suspend request is active (FCNFG[ERSSUSP]=1).

FTFA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0000	Flash Status Register (FTFA_FSTAT)	8	R/W	00h	23.33.1/387
4002_0001	Flash Configuration Register (FTFA_FCNFG)	8	R/W	00h	23.33.2/388
4002_0002	Flash Security Register (FTFA_FSEC)	8	R	Undefined	23.33.3/390
4002_0003	Flash Option Register (FTFA_FOPT)	8	R	Undefined	23.33.4/391
4002_0004	Flash Common Command Object Registers (FTFA_FCCOB3)	8	R/W	00h	23.33.5/392
4002_0005	Flash Common Command Object Registers (FTFA_FCCOB2)	8	R/W	00h	23.33.5/392
4002_0006	Flash Common Command Object Registers (FTFA_FCCOB1)	8	R/W	00h	23.33.5/392
4002_0007	Flash Common Command Object Registers (FTFA_FCCOB0)	8	R/W	00h	23.33.5/392
4002_0008	Flash Common Command Object Registers (FTFA_FCCOB7)	8	R/W	00h	23.33.5/392
4002_0009	Flash Common Command Object Registers (FTFA_FCCOB6)	8	R/W	00h	23.33.5/392

Table continues on the next page...

FTFA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_000A	Flash Common Command Object Registers (FTFA_FCCOB5)	8	R/W	00h	23.33.5/392
4002_000B	Flash Common Command Object Registers (FTFA_FCCOB4)	8	R/W	00h	23.33.5/392
4002_000C	Flash Common Command Object Registers (FTFA_FCCOBB)	8	R/W	00h	23.33.5/392
4002_000D	Flash Common Command Object Registers (FTFA_FCCOBA)	8	R/W	00h	23.33.5/392
4002_000E	Flash Common Command Object Registers (FTFA_FCCOB9)	8	R/W	00h	23.33.5/392
4002_000F	Flash Common Command Object Registers (FTFA_FCCOB8)	8	R/W	00h	23.33.5/392
4002_0010	Program Flash Protection Registers (FTFA_FPROT3)	8	R/W	Undefined	23.33.6/393
4002_0011	Program Flash Protection Registers (FTFA_FPROT2)	8	R/W	Undefined	23.33.6/393
4002_0012	Program Flash Protection Registers (FTFA_FPROT1)	8	R/W	Undefined	23.33.6/393
4002_0013	Program Flash Protection Registers (FTFA_FPROT0)	8	R/W	Undefined	23.33.6/393

23.33.1 Flash Status Register (FTFA_FSTAT)

The FSTAT register reports the operational status of the flash memory module.

The CCIF, RDCOLERR, ACCERR, and FPVIOL bits are readable and writable. The MGSTAT0 bit is read only. The unassigned bits read 0 and are not writable.

NOTE

When set, the Access Error (ACCERR) and Flash Protection Violation (FPVIOL) bits in this register prevent the launch of any more commands until the flag is cleared (by writing a one to it).

Address: 4002_0000h base + 0h offset = 4002_0000h

Bit	7	6	5	4	3	2	1	0
Read	CCIF	RDCOLERR	ACCERR	FPVIOL	0			MGSTAT0
Write	w1c	w1c	w1c	w1c				
Reset	0	0	0	0	0	0	0	0

FTFA_FSTAT field descriptions

Field	Description
7 CCIF	Command Complete Interrupt Flag

Table continues on the next page...

FTFA_FSTAT field descriptions (continued)

Field	Description
	Indicates that a flash command has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command, and CCIF stays low until command completion or command violation. CCIF is reset to 0 but is set to 1 by the memory controller at the end of the reset initialization sequence. Depending on how quickly the read occurs after reset release, the user may or may not see the 0 hardware reset value. 0 Flash command in progress 1 Flash command has completed
6 RDCOLERR	Flash Read Collision Error Flag Indicates that the MCU attempted a read from a flash memory resource that was being manipulated by a flash command (CCIF=0). Any simultaneous access is detected as a collision error by the block arbitration logic. The read data in this case cannot be guaranteed. The RDCOLERR bit is cleared by writing a 1 to it. Writing a 0 to RDCOLERR has no effect. 0 No collision error detected 1 Collision error detected
5 ACCERR	Flash Access Error Flag Indicates an illegal access has occurred to a flash memory resource caused by a violation of the command write sequence or issuing an illegal flash command. While ACCERR is set, the CCIF flag cannot be cleared to launch a command. The ACCERR bit is cleared by writing a 1 to it. Writing a 0 to the ACCERR bit has no effect. 0 No access error detected 1 Access error detected
4 FPVIOL	Flash Protection Violation Flag Indicates an attempt was made to program or erase an address in a protected area of program flash memory during a command write sequence . While FPVIOL is set, the CCIF flag cannot be cleared to launch a command. The FPVIOL bit is cleared by writing a 1 to it. 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 a flash command or during the flash reset sequence. As a status flag, this field cannot (and need not) be cleared by the user like the other error flags in this register. The value of the MGSTAT0 bit for "command-N" is valid only at the end of the "command-N" execution when CCIF=1 and before the next command has been launched. At some point during the execution of "command-N+1," the previous result is discarded and any previous error is cleared.

23.33.2 Flash Configuration Register (FTFA_FCNFG)

This register provides information on the current functional state of the flash memory module.

The erase control bits (ERSAREQ and ERSSUSP) have write restrictions. The unassigned bits read as noted and are not writable.

Address: 4002_0000h base + 1h offset = 4002_0001h

Bit	7	6	5	4	3	2	1	0
Read	CCIE	RDCOLLIE	ERSAREQ	ERSSUSP	0	0	0	0
Write								
Reset	0	0	0	0	0	0	0	0

FTFA_FCNFG field descriptions

Field	Description
7 CCIE	Command Complete Interrupt Enable Controls interrupt generation when a flash command completes. 0 Command complete interrupt disabled 1 Command complete interrupt enabled. An interrupt request is generated whenever the FSTAT[CCIF] flag is set.
6 RDCOLLIE	Read Collision Error Interrupt Enable Controls interrupt generation when a flash memory read collision error occurs. 0 Read collision error interrupt disabled 1 Read collision error interrupt enabled. An interrupt request is generated whenever a flash memory read collision error is detected (see the description of FSTAT[RDCOLERR]).
5 ERSAREQ	Erase All Request Issues a request to the memory controller to execute the Erase All Blocks command and release security. ERSAREQ is not directly writable but is under indirect user control. Refer to the device's Chip Configuration details on how to request this command. ERSAREQ sets when an erase all request is triggered external to the flash memory module and CCIF is set (no command is currently being executed). ERSAREQ is cleared by the flash memory module when the operation completes. 0 No request or request complete 1 Request to: 1. run the Erase All Blocks command, 2. verify the erased state, 3. program the security byte in the Flash Configuration Field to the unsecure state, and 4. release MCU security by setting the FSEC[SEC] field to the unsecure state.
4 ERSSUSP	Erase Suspend Allows the user to suspend (interrupt) the Erase Flash Sector command while it is executing. 0 No suspend requested 1 Suspend the current Erase Flash Sector command execution.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

FTFA_FCNFG field descriptions (continued)

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

23.33.3 Flash Security Register (FTFA_FSEC)

This read-only register holds all bits associated with the security of the MCU and flash memory module.

During the reset sequence, the register is loaded with the contents of the flash security byte in the Flash Configuration Field located in program flash memory. The flash basis for the values is signified by X in the reset value.

Address: 4002_0000h base + 2h offset = 4002_0002h

Bit	7	6	5	4	3	2	1	0
Read	KEYEN		MEEN		FSLACC		SEC	
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFA_FSEC field descriptions

Field	Description
7–6 KEYEN	Backdoor Key Security Enable Enables or disables backdoor key access to the flash memory module. 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 flash memory module. The state of this field is relevant only when SEC is set to secure outside of NVM Normal Mode. When SEC is set to unsecure, the MEEN setting does not matter. 00 Mass erase is enabled 01 Mass erase is enabled 10 Mass erase is disabled 11 Mass erase is enabled
3–2 FSLACC	Freescale Failure Analysis Access Code

Table continues on the next page...

FTFA_FSEC field descriptions (continued)

Field	Description
	Enables or disables access to the flash memory contents during returned part failure analysis at Freescale. When SEC is secure and FSLACC is denied, access to the program flash contents is denied and any failure analysis performed by Freescale factory test must begin with a full erase to unsecure the part. When access is granted (SEC is unsecure, or SEC is secure and FSLACC is granted), Freescale factory testing has visibility of the current flash contents. The state of the FSLACC bits is only relevant when SEC is set to secure. When SEC is set to unsecure, the FSLACC setting does not matter. 00 Freescale factory access granted 01 Freescale factory access denied 10 Freescale factory access denied 11 Freescale factory access granted
1–0 SEC	Flash Security Defines the security state of the MCU. In the secure state, the MCU limits access to flash memory module resources. The limitations are defined per device and are detailed in the Chip Configuration details. If the flash memory module is unsecured using backdoor key access, SEC is forced to 10b. 00 MCU security status is secure. 01 MCU security status is secure. 10 MCU security status is unsecure. (The standard shipping condition of the flash memory module is unsecure.) 11 MCU security status is secure.

23.33.4 Flash Option Register (FTFA_FOPT)

The flash option register allows the MCU to customize its operations by examining the state of these read-only bits, which are loaded from NVM at reset. The function of the bits is defined in the device's Chip Configuration details.

All bits in the register are read-only .

During the reset sequence, the register is loaded from the flash nonvolatile option byte in the Flash Configuration Field located in program flash memory. The flash basis for the values is signified by X in the reset value.

Address: 4002_0000h base + 3h offset = 4002_0003h

Bit	7	6	5	4	3	2	1	0
Read	OPT							
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFA_FOPT field descriptions

Field	Description
7–0 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.

23.33.5 Flash Common Command Object Registers (FTFA_FCCOB*n*)

The FCCOB register group provides 12 bytes for command codes and parameters. The individual bytes within the set append a 0-B hex identifier to the FCCOB register name: FCCOB0, FCCOB1, ..., FCCOBB.

Address: 4002_0000h base + 4h offset + (1d × i), where i=0d to 11d

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

FTFA_FCCOB*n* field descriptions

Field	Description
7–0 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. The following table shows a generic flash command format. The first FCCOB register, FCCOB0, always contains the command code. This 8-bit value defines the command to be executed. The command code is followed by the parameters required for this specific flash command, typically an address and/or data values. **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	Typical Command Parameter Contents [7:0]
0	FCMD (a code that defines the flash command)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0]

FTFA_FCCOB*n* field descriptions (continued)

Field	Description
	(nested table below)

FCCOB Number	Typical Command Parameter Contents [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).

23.33.6 Program Flash Protection Registers (FTFA_FPROT*n*)

The FPROT registers define which program flash regions are protected from program and erase operations. Protected flash regions cannot have their content changed; that is, these regions cannot be programmed and cannot be erased by any flash command. Unprotected regions can be changed by program and erase operations.

The four FPROT registers allow up to 32 protectable regions. Each bit protects a 1/32 region of the program flash memory except for memory configurations with less than 32 KB of program flash where each assigned bit protects 1 KB . For configurations with 24 KB of program flash memory or less, FPROT0 is not used. For configurations with 16 KB of program flash memory or less, FPROT1 is not used. For configurations with 8 KB of program flash memory, FPROT2 is not used. The bitfields are defined in each register as follows:

Program flash protection register	Program flash protection bits
FPROT0	PROT[31:24]
FPROT1	PROT[23:16]
FPROT2	PROT[15:8]
FPROT3	PROT[7:0]

During the reset sequence, the FPROT registers are loaded with the contents of the program flash protection bytes in the Flash Configuration Field as indicated in the following table.

Program flash protection register	Flash Configuration Field offset address
FPROT0	0x000B
FPROT1	0x000A
FPROT2	0x0009
FPROT3	0x0008

To change the program flash protection that is loaded during the reset sequence, unprotect the sector of program flash memory that contains the Flash Configuration Field. Then, reprogram the program flash protection byte.

Address: 4002_0000h base + 10h offset + (1d × i), where i=0d to 3d

Bit	7	6	5	4	3	2	1	0
Read Write	PROT							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFA_FPROT*n* field descriptions

Field	Description
7–0 PROT	Program Flash Region Protect Each program flash region can be protected from program and erase operations by setting the associated PROT 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 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. Each bit in the 32-bit protection register represents 1/32 of the total program flash except for configurations where program flash memory is less than 32 KB. For configurations with less than 32 KB of program flash memory, each assigned bit represents 1 KB. 0 Program flash region is protected. 1 Program flash region is not protected

23.4 Functional Description

The information found here describes functional details of the flash memory module.

23.4.1 Flash Protection

Individual regions within the flash memory can be protected from program and erase operations.

Protection is controlled by the following registers:

- FPROT*n* —
  - For 2^n program flash sizes, four registers typically protect 32 regions of the program flash memory as shown in the following figure

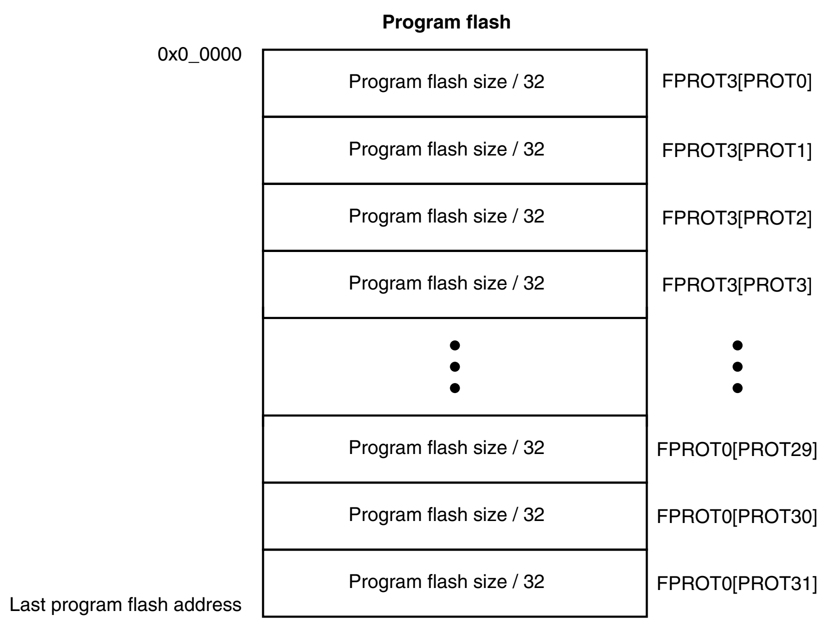


Figure 23-24. Program flash protection

NOTE

Flash protection features are discussed further in AN4507: Using the Kinetis Security and Flash Protection Features . Not all features described in the application note are available on this device.

23.4.2 Interrupts

The flash memory module can generate interrupt requests to the MCU upon the occurrence of various flash events.

These interrupt events and their associated status and control bits are shown in the following table.

Table 23-24. Flash Interrupt Sources

Flash Event	Readable Status Bit	Interrupt Enable Bit
Flash Command Complete	FSTAT[CCIF]	FCNFG[CCIE]
Flash Read Collision Error	FSTAT[RDCOLERR]	FCNFG[RDCOLLIE]

Note

Vector addresses and their relative interrupt priority are determined at the MCU level.

Some devices also generate a bus error response as a result of a Read Collision Error event. See the chip configuration information to determine if a bus error response is also supported.

23.4.3 Flash Operation in Low-Power Modes

23.4.3.1 Wait Mode

When the MCU enters wait mode, the flash memory module is not affected. The flash memory module can recover the MCU from wait via the command complete interrupt (see Interrupts).

23.4.3.2 Stop Mode

When the MCU requests stop mode, if a flash command is active (CCIF = 0) the command execution completes before the MCU is allowed to enter stop mode.

CAUTION

The MCU should never enter stop mode while any flash command is running (CCIF = 0).

NOTE

While the MCU is in very-low-power modes (VLPR, VLPW, VLPS), the flash memory module does not accept flash commands.

23.4.4 Functional Modes of Operation

The flash memory module has two operating modes: NVM Normal and NVM Special.

The operating mode affects the command set availability (see Table 23-25). Refer to the Chip Configuration details of this device for how to activate each mode.

23.4.5 Flash Reads and Ignored Writes

The flash memory module requires only the flash address to execute a flash memory read.

The MCU must not read from the flash memory while commands are running (as evidenced by CCIF=0) on that block. Read data cannot be guaranteed from a flash block while any command is processing within that block. The block arbitration logic detects any simultaneous access and reports this as a read collision error (see the FSTAT[RDCOLERR] bit).

23.4.6 Read While Write (RWW)

The following simultaneous accesses are not allowed:

- Reading from program flash memory space while a flash command is active (CCIF=0).

23.4.7 Flash Program and Erase

All flash functions except read require the user to setup and launch a flash command through a series of peripheral bus writes.

The user cannot initiate any further flash commands until notified that the current command has completed. The flash command structure and operation are detailed in Flash Command Operations.

23.4.8 Flash Command Operations

Flash command operations are typically used to modify flash memory contents.

The next sections describe:

- The command write sequence used to set flash command parameters and launch execution
- A description of all flash commands available

23.4.8.1 Command Write Sequence

Flash commands are specified using a command write sequence illustrated in Figure 23-25. The flash memory module performs various checks on the command (FCCOB) content and continues with command execution if all requirements are fulfilled.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be zero and the CCIF flag must read 1 to verify that any previous command has completed. If CCIF is zero, the previous command execution is still active, a new command write sequence cannot be started, and all writes to the FCCOB registers are ignored.

Attempts to launch a flash command in VLP mode will be ignored.

23.4.8.1.1 Load the FCCOB Registers

The user must load the FCCOB registers with all parameters required by the desired flash command. The individual registers that make up the FCCOB data set can be written in any order.

23.4.8.1.2 Launch the Command by Clearing CCIF

Once all relevant command parameters have been loaded, the user launches the command by clearing FSTAT[CCIF] by writing a '1' to it. FSTAT[CCIF] remains 0 until the flash command completes.

The FSTAT register contains a blocking mechanism that prevents a new command from launching (can't clear FSTAT[CCIF]) if the previous command resulted in an access error (FSTAT[ACCERR]=1) or a protection violation (FSTAT[FPVIOL]=1). In error scenarios, two writes to FSTAT are required to initiate the next command: the first write clears the error flags, the second write clears CCIF.

23.4.8.1.3 Command Execution and Error Reporting

The command processing has several steps:

1. The flash memory module reads the command code and performs a series of parameter checks and protection checks, if applicable, which are unique to each command.

   If the parameter check fails, the FSTAT[ACCERR] (access error) flag is set. FSTAT[ACCERR] reports invalid instruction codes and out-of bounds addresses. Usually, access errors suggest that the command was not set-up with valid parameters in the FCCOB register group.

   Program and erase commands also check the address to determine if the operation is requested to execute on protected areas. If the protection check fails, FSTAT[FPVIOL] (protection error) flag is set.

   Command processing never proceeds to execution when the parameter or protection step fails. Instead, command processing is terminated after setting FSTAT[CCIF].

2. If the parameter and protection checks pass, the command proceeds to execution. Run-time errors, such as failure to erase verify, may occur during the execution phase. Run-time errors are reported in FSTAT[MGSTAT0]. A command may have access errors, protection errors, and run-time errors, but the run-time errors are not seen until all access and protection errors have been corrected.
3. Command execution results, if applicable, are reported back to the user via the FCCOB and FSTAT registers.
4. The flash memory module sets FSTAT[CCIF] signifying that the command has completed.

The flow for a generic command write sequence is illustrated in the following figure.

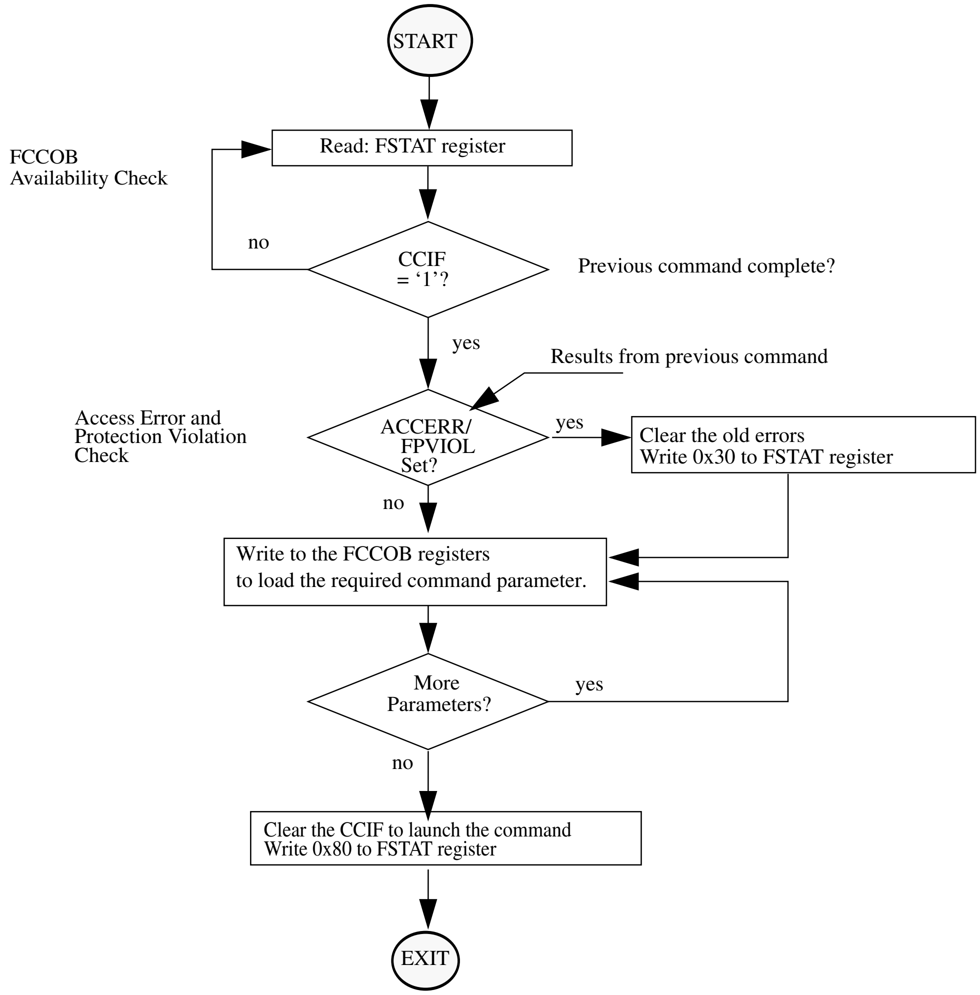


Figure 23-25. Generic flash command write sequence flowchart

23.4.8.2 Flash Commands

The following table summarizes the function of all flash commands.

FCMD	Command	Program flash	Function
0x01	Read 1s Section	×	Verify that a given number of program flash locations from a starting address are erased.
0x02	Program Check	×	Tests previously-programmed locations at margin read levels.
0x03	Read Resource	IFR, ID	Read 4 bytes from program flash IFR or version ID.
0x06	Program Longword	×	Program 4 bytes in a program flash block.

Table continues on the next page...

FCMD	Command	Program flash	Function
0x09	Erase Flash Sector	×	Erase all bytes in a program flash sector.
0x40	Read 1s All Blocks	×	Verify that the program flash block is erased then release MCU security.
0x41	Read Once	IFR	Read 4 bytes of a dedicated 64 byte field in the program flash 0 IFR.
0x43	Program Once	IFR	One-time program of 4 bytes of a dedicated 64-byte field in the program flash 0 IFR.
0x44	Erase All Blocks	×	Erase the program flash block, 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.

23.4.8.3 Flash Commands by Mode

The following table shows the flash commands that can be executed in each flash operating mode.

Table 23-25. Flash Commands by Mode

FCMD	Command	NVM Normal			NVM Special		
		Unsecure	Secure	MEEN=10	Unsecure	Secure	MEEN=10
0x01	Read 1s Section	×	×	×	×	—	—
0x02	Program Check	×	×	×	×	—	—
0x03	Read Resource	×	×	×	×	—	—
0x06	Program Longword	×	×	×	×	—	—
0x09	Erase Flash Sector	×	×	×	×	—	—
0x40	Read 1s All Blocks	×	×	×	×	×	—
0x41	Read Once	×	×	×	×	—	—
0x43	Program Once	×	×	×	×	—	—
0x44	Erase All Blocks	×	×	×	×	×	—
0x45	Verify Backdoor Access Key	×	×	×	×	—	—

23.4.9 Margin Read Commands

The Read-1s commands (Read 1s All Blocks and Read 1s Section) and the Program Check command have a margin choice parameter that allows the user to apply non-standard read reference levels to the program flash array reads performed by these commands. Using the preset 'user' and 'factory' margin levels, these commands perform their associated read operations at tighter tolerances than a 'normal' read. These non-standard read levels are applied only during the command execution. All simple (uncommanded) flash array reads to the MCU always use the standard, un-margined, read reference level.

Only the 'normal' read level should be employed during normal flash usage. The non-standard, 'user' and 'factory' margin levels should be employed only in special cases. They can be used during special diagnostic routines to gain confidence that the device is not suffering from the end-of-life data loss customary of flash memory devices.

Erased ('1') and programmed ('0') bit states can degrade due to elapsed time and data cycling (number of times a bit is erased and re-programmed). The lifetime of the erased states is relative to the last erase operation. The lifetime of the programmed states is measured from the last program time.

The 'user' and 'factory' levels become, in effect, a minimum safety margin; i.e. if the reads pass at the tighter tolerances of the 'user' and 'factory' margins, then the 'normal' reads have at least this much safety margin before they experience data loss.

The 'user' margin is a small delta to the normal read reference level. 'User' margin levels can be employed to check that flash memory contents have adequate margin for normal level read operations. If unexpected read results are encountered when checking flash memory contents at the 'user' margin levels, loss of information might soon occur during 'normal' readout.

The 'factory' margin is a bigger deviation from the norm, a more stringent read criteria that should only be attempted immediately (or very soon) after completion of an erase or program command, early in the cycling life. 'Factory' margin levels can be used to check that flash memory contents have adequate margin for long-term data retention at the normal level setting. If unexpected results are encountered when checking flash memory contents at 'factory' margin levels, the flash memory contents should be erased and reprogrammed.

CAUTION

Factory margin levels must only be used during verify of the initial factory programming.

23.4.10 Flash Command Description

This section describes all flash commands that can be launched by a command write sequence.

The flash memory module sets the FSTAT[ACCERR] bit and aborts the command execution if any of the following illegal conditions occur:

- There is an unrecognized command code in the FCCOB FCMD field.
- There is an error in a FCCOB field for the specific commands. Refer to the error handling table provided for each command.

Ensure that FSTAT[ACCERR] and FSTAT[FPVIOL] are cleared prior to starting the command write sequence. As described in Launch the Command by Clearing CCIF, a new command cannot be launched while these error flags are set.

Do not attempt to read a flash block while the flash memory module is running a command (FSTAT[CCIF] = 0) on that same block. The flash memory module may return invalid data to the MCU with the collision error flag (FSTAT[RDCOLERR]) set.

CAUTION

Flash data must be in the erased state before being programmed. Cumulative programming of bits (adding more zeros) is not allowed.

23.4.10.1 Read 1s Section Command

The Read 1s Section command checks if a section of program flash memory is erased to the specified read margin level. The Read 1s Section command defines the starting address and the number of longwords to be verified.

Table 23-26. Read 1s Section Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x01 (RD1SEC)
1	Flash address [23:16] of the first longword to be verified
2	Flash address [15:8] of the first longword to be verified
3	Flash address [7:0][1] of the first longword to be verified
4	Number of longwords to be verified [15:8]
5	Number of longwords to be verified [7:0]
6	Read-1 Margin Choice

1. Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Read 1s Section command, the flash memory module sets the read margin for 1s according to Table 23-27 and then reads all locations within the specified section of flash memory. If the flash memory module fails to read all 1s (that is, the flash section is not erased), FSTAT[MGSTAT0] is set. FSTAT[CCIF] sets after the Read 1s Section operation completes.

Table 23-27. 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 23-28. 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 longword aligned.	FSTAT[ACCERR]
The requested section crosses a Flash block boundary.	FSTAT[ACCERR]
The requested number of longwords is 0.	FSTAT[ACCERR]
Read-1s fails.	FSTAT[MGSTAT0]

23.4.10.2 Program Check Command

The Program Check command tests a previously programmed program flash longword to see if it reads correctly at the specified margin level.

Table 23-29. Program Check Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x02 (PGMCHK)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Margin Choice
8	Byte 0 expected data
9	Byte 1 expected data
A	Byte 2 expected data
B	Byte 3 expected data

1. Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Program Check command, the flash memory module sets the read margin for 1s according to Table 23-30, reads the specified longword, and compares the actual read data to the expected data provided by the FCCOB. If the comparison at margin-1 fails, FSTAT[MGSTAT0] is set.

The flash memory module then sets the read margin for 0s, re-reads, and compares again. If the comparison at margin-0 fails, FSTAT[MGSTAT0] is set. FSTAT[CCIF] is set after the Program Check operation completes.

The supplied address must be longword aligned (the lowest two bits of the byte address must be 00):

- Byte 3 data is written to the supplied byte address ('start'),
- Byte 2 data is programmed to byte address start+0b01,
- Byte 1 data is programmed to byte address start+0b10,
- Byte 0 data is programmed to byte address start+0b11.

NOTE

See the description of margin reads, Margin Read Commands

Table 23-30. 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 23-31. 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]

23.4.10.3 Read Resource Command

The Read Resource command allows the user to read data from special-purpose memory resources located within the flash memory module. The special-purpose memory resources available include program flash IFR space and the Version ID field. Each resource is assigned a select code as shown in Table 23-33.

Table 23-32. Read Resource Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x03 (RDRSRC)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
Returned Values	
4	Read Data [31:24]
5	Read Data [23:16]
6	Read Data [15:8]
7	Read Data [7:0]
User-provided values	
8	Resource Select Code (see Table 23-33)

1. Must be longword aligned (Flash address [1:0] = 00).

Table 23-33. Read Resource Select Codes

Resource Select Code	Description	Resource Size	Local Address Range
0x00	Program Flash 0 IFR	256 Bytes	0x00_0000–0x00_00FF
0x01[1]	Version ID	8 Bytes	0x00_0000–0x00_0007

1. Located in program flash 0 reserved space.

After clearing CCIF to launch the Read Resource command, four consecutive bytes are read from the selected resource at the provided relative address and stored in the FCCOB register. The CCIF flag sets after the Read Resource operation completes. The Read Resource command exits with an access error if an invalid resource code is provided or if the address for the applicable area is out-of-range.

Table 23-34. Read Resource Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid resource code is entered	FSTAT[ACCERR]
Flash address is out-of-range for the targeted resource.	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]

23.4.10.4 Program Longword Command

The Program Longword command programs four previously-erased bytes in the program flash memory using an embedded algorithm.

CAUTION

A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash memory location is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.

Table 23-35. Program Longword Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x06 (PGM4)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Byte 0 program value
5	Byte 1 program value
6	Byte 2 program value
7	Byte 3 program value

1. Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Program Longword command, the flash memory module programs the data bytes into the flash using the supplied address. The targeted flash locations must be currently unprotected (see the description of the FPROT registers) to permit execution of the Program Longword operation.

The programming operation is unidirectional. It can only move NVM bits from the erased state ('1') to the programmed state ('0'). Erased bits that fail to program to the '0' state are flagged as errors in FSTAT[MGSTAT0]. The CCIF flag is set after the Program Longword operation completes.

The supplied address must be longword aligned (flash address [1:0] = 00):

- Byte 3 data is written to the supplied byte address ('start'),
- Byte 2 data is programmed to byte address start+0b01,
- Byte 1 data is programmed to byte address start+0b10, and
- Byte 0 data is programmed to byte address start+0b11.

Table 23-36. Program Longword Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid flash address is supplied	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
Flash address points to a protected area	FSTAT[FPVIOL]

Table continues on the next page...

Table 23-36. Program Longword Command Error Handling (continued)

Error Condition	Error Bit
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

23.4.10.5 Erase Flash Sector Command

The Erase Flash Sector operation erases all addresses in a flash sector.

Table 23-37. 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 longword aligned (flash address [1:0] = 00).

After clearing CCIF to launch the Erase Flash Sector command, the flash memory module erases the selected program flash sector and then verifies that it is erased. The Erase Flash Sector command aborts if the selected sector is protected (see the description of the FPROT registers). If the erase-verify fails the FSTAT[MGSTAT0] bit is set. The CCIF flag is set after the Erase Flash Sector operation completes. The Erase Flash Sector command is suspendable (see the FCNFG[ERSSUSP] bit and Figure 23-26).

Table 23-38. 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 longword aligned	FSTAT[ACCERR]
The selected program 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.

23.4.10.5.1 Suspending an Erase Flash Sector Operation

To suspend an Erase Flash Sector operation set the FCNFG[ERSSUSP] bit (see Flash Configuration Field Description) when CCIF is clear and the CCOB command field holds the code for the Erase Flash Sector command. During the Erase Flash Sector operation (see Erase Flash Sector Command), the flash memory module samples the state of the ERSSUSP bit at convenient points. If the flash memory module detects that the

ERSSUSP bit is set, the Erase Flash Sector operation is suspended and the flash memory module sets CCIF. While ERSSUSP is set, all writes to flash registers are ignored except for writes to the FSTAT and FCNFG registers.

If an Erase Flash Sector operation effectively completes before the flash memory module detects that a suspend request has been made, the flash memory module clears the ERSSUSP bit prior to setting CCIF. When an Erase Flash Sector operation has been successfully suspended, the flash memory module sets CCIF and leaves the ERSSUSP bit set. While CCIF is set, the ERSSUSP bit can only be cleared to prevent the withdrawal of a suspend request before the flash memory module has acknowledged it.

23.4.10.5.2 Resuming a Suspended Erase Flash Sector Operation

If the ERSSUSP bit is still set when CCIF is cleared to launch the next command, the previous Erase Flash Sector operation resumes. The flash memory module acknowledges the request to resume a suspended operation by clearing the ERSSUSP bit. A new suspend request can then be made by setting ERSSUSP. A single Erase Flash Sector operation can be suspended and resumed multiple times.

There is a minimum elapsed time limit 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.

23.4.10.5.3 Aborting a Suspended Erase Flash Sector Operation

The user may choose to abort a suspended Erase Flash Sector operation by clearing the ERSSUSP bit prior to clearing CCIF for the next command launch. When a suspended operation is aborted, the flash memory module starts the new command using the new FCCOB contents.

Note

> Aborting the erase leaves the bitcells in an indeterminate, partially-erased state. Data in this sector is not reliable until a new erase command fully completes.

The following figure shows how to suspend and resume the Erase Flash Sector operation.

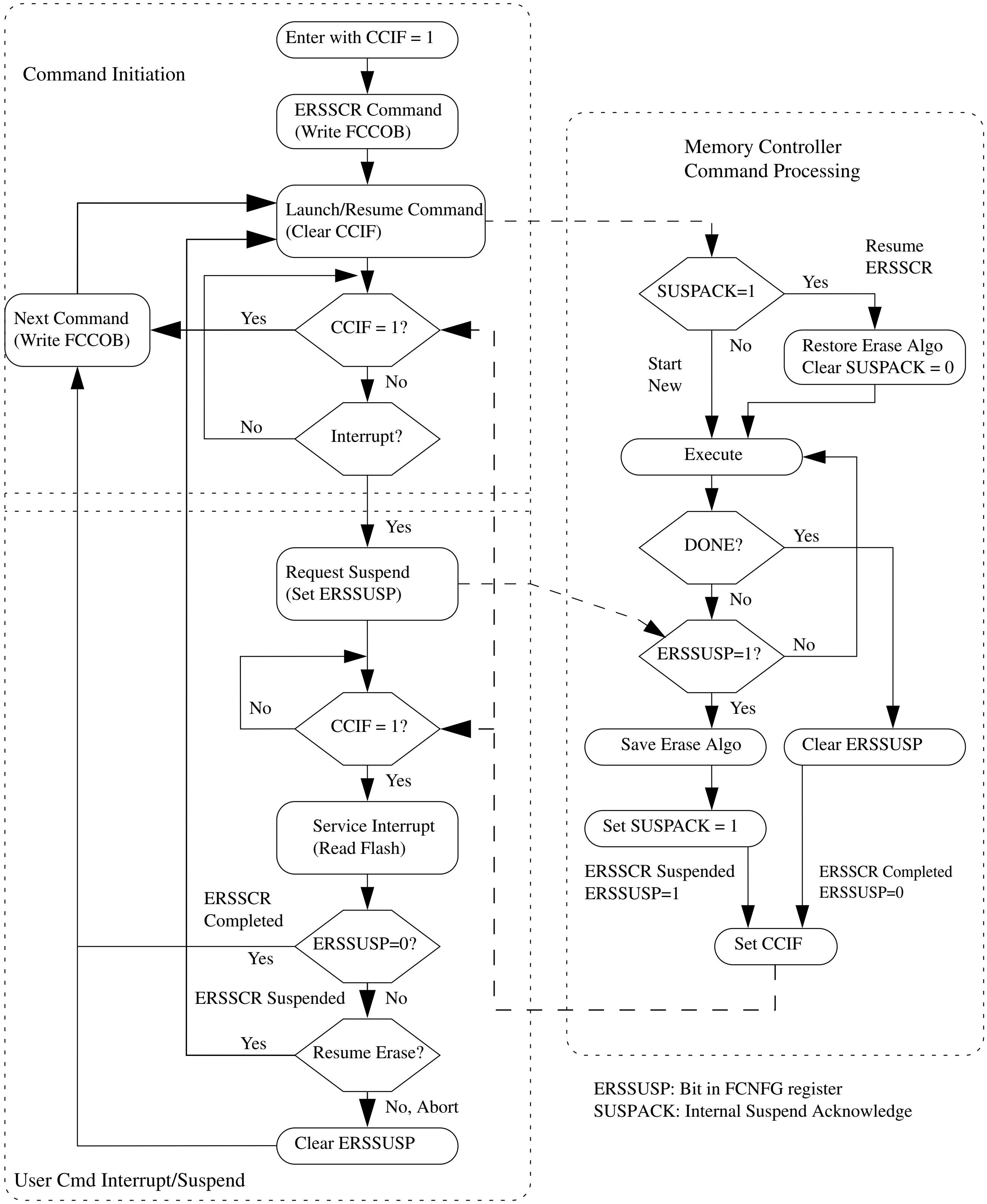


Figure 23-26. Suspend and Resume of Erase Flash Sector Operation

23.4.10.6 Read 1s All Blocks Command

The Read 1s All Blocks command checks if the program flash blocks have been erased to the specified read margin level, if applicable, and releases security if the readout passes, i.e. all data reads as '1'.

Table 23-39. Read 1s All Blocks Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x40 (RD1ALL)
1	Read-1 Margin Choice

After clearing CCIF to launch the Read 1s All Blocks command, the flash memory module :

- sets the read margin for 1s according to Table 23-40,
- checks the contents of the program flash are in the erased state.

If the flash memory module 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 memory resources are not in the fully erased state, the FSTAT[MGSTAT0] bit is set.

The CCIF flag sets after the Read 1s All Blocks operation has completed.

Table 23-40. 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 23-41. Read 1s All Blocks Command Error Handling

Error Condition	Error Bit
An invalid margin choice is specified	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

23.4.10.7 Read Once Command

The Read Once command provides read access to a reserved 64-byte field located in the program flash 0 IFR (see Program Flash IFR Map and Program Once Field). Access to the Program Once field is via 16 records, each 4 bytes long. The Program Once field is programmed using the Program Once command described in Program Once Command.

Table 23-42. Read Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x41 (RDONCE)
1	Program Once record index (0x00 - 0x0F)
2	Not used
3	Not used
Returned Values	
4	Program Once byte 0 value
5	Program Once byte 1 value
6	Program Once byte 2 value
7	Program Once byte 3 value

After clearing CCIF to launch the Read Once command, a 4-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 0x0F. 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 23-43. 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]

23.4.10.8 Program Once Command

The Program Once command enables programming to a reserved 64-byte field in the program flash 0 IFR (see Program Flash IFR Map and Program Once Field). Access to the Program Once field is via 16 records, each 4 bytes long. The Program Once field can be read using the Read Once command (see Read Once Command) or using the Read Resource command (see Read Resource Command). Each Program Once record can be programmed only once since the program flash 0 IFR cannot be erased.

Table 23-44. Program Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x43 (PGMONCE)
1	Program Once record index (0x00 - 0x0F)
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

After clearing CCIF to launch the Program Once command, the flash memory module first verifies that the selected record is erased. If erased, then the selected record is programmed using the values provided. The Program Once command also verifies that the programmed values read back correctly. The CCIF flag is set after the Program Once operation has completed.

The reserved program flash 0 IFR location accessed by the Program Once command cannot be erased and any attempt to program one of these records when the existing value is not Fs (erased) is not allowed. Valid record index values for the Program Once command range from 0x00 to 0x0F. During execution of the Program Once command, any attempt to read addresses within the program flash block containing this 64-byte field returns invalid data.

Table 23-45. Program Once Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid record index is supplied	FSTAT[ACCERR]
The requested record has already been programmed to a non-FFFF value[1]	FSTAT[ACCERR]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

1. If a Program Once record is initially programmed to 0xFFFF_FFFF, the Program Once command is allowed to execute again on that same record.

23.4.10.9 Erase All Blocks Command

The Erase All Blocks operation erases all flash memory, verifies all memory contents, and releases MCU security.

Table 23-46. Erase All Blocks Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x44 (ERSALL)

After clearing CCIF to launch the Erase All Blocks command, the flash memory module erases all program flash memory, then verifies that all are erased.

If the flash memory module verifies that all flash memories were properly erased, security is released by setting the FSEC[SEC] field to the unsecure state. The Erase All Blocks command aborts if any flash region is protected. The security byte and all other contents of the flash configuration field (see Flash Configuration Field Description) are erased by the Erase All Blocks command. If the erase-verify fails, the FSTAT[MGSTAT0] bit is set. The CCIF flag is set after the Erase All Blocks operation completes.

Table 23-47. Erase All Blocks Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
Any region of the program flash memory is protected	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation[1]	FSTAT[MGSTAT0]

1. User margin read may be run using the Read 1s All Blocks command to verify all bits are erased.

23.4.10.9.1 Triggering an Erase All External to the Flash Memory Module

The functionality of the Erase All BlocksErase 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 FSTAT[ACCERR and PVIOL] flags must be cleared and the FCCOB0 register must not contain 0x44. When invoked, the erase-all function erases all program flash memory regardless of the protection settings. If the post-erase verify passes, the routine then releases security by setting the FSEC[SEC] field register to the unsecure state. The security byte in the Flash Configuration Field is also programmed to the unsecure state. The status of the erase-all request is reflected in the FCNFG[ERSAREQ] bit. The FCNFG[ERSAREQ] bit is cleared once the operation completes and the normal FSTAT error reporting is available as described in Erase All Blocks Command.

23.4.10.10 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command only executes if the mode and security conditions are satisfied (see Flash Commands by Mode). Execution of the Verify Backdoor Access Key command is further qualified by the FSEC[KEYEN] bits. The Verify Backdoor Access Key command releases security if user-supplied keys in the FCCOB match those stored in the Backdoor Comparison Key bytes of the Flash Configuration Field (see Flash Configuration Field Description). The column labelled Flash Configuration Field offset address shows the location of the matching byte in the Flash Configuration Field.

Table 23-48. 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_0000
5	Key Byte 1	0x0_0001
6	Key Byte 2	0x0_0002
7	Key Byte 3	0x0_0003
8	Key Byte 4	0x0_0004
9	Key Byte 5	0x0_0005
A	Key Byte 6	0x0_0006
B	Key Byte 7	0x0_0007

After clearing CCIF to launch the Verify Backdoor Access Key command, the flash memory module checks the FSEC[KEYEN] bits to verify that this command is enabled. If not enabled, the flash memory module sets the FSTAT[ACCERR] bit and terminates. If the command is enabled, the flash memory module compares the key provided in FCCOB to the backdoor comparison key in the Flash Configuration Field. If the backdoor keys match, the FSEC[SEC] field is changed to the unsecure state and security is released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are immediately aborted and the FSTAT[ACCERR] bit is (again) set to 1 until a reset of the flash memory module module occurs. If the entire 8-byte key is all zeros or all ones, the Verify Backdoor Access Key command fails with an access error. The CCIF flag is set after the Verify Backdoor Access Key operation completes.

Table 23-49. Verify Backdoor Access Key Command Error Handling

Error Condition	Error Bit
The supplied key is all-0s or all-Fs	FSTAT[ACCERR]
An incorrect backdoor key is supplied	FSTAT[ACCERR]

Table continues on the next page...

Table 23-49. Verify Backdoor Access Key Command Error Handling (continued)

Error Condition	Error Bit
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]

23.4.11 Security

The flash memory module provides security information to the MCU based on contents of the FSEC security register.

The MCU then limits access to flash memory resources as defined in the device's Chip Configuration details. During reset, the flash memory module initializes the FSEC register using data read from the security byte of the Flash Configuration Field (see Flash Configuration Field Description).

The following fields are available in the FSEC register. The settings are described in the Flash Security Register (FTFA_FSEC) details.

Flash security features are discussed further in AN4507: Using the Kinetis Security and Flash Protection Features . Note that not all features described in the application note are available on this device.

Table 23-50. FSEC register fields

FSEC field	Description
KEYEN	Backdoor Key Access
MEEN	Mass Erase Capability
FSLACC	Freescale Factory Access
SEC	MCU security

23.4.11.1 Flash Memory Access by Mode and Security

The following table summarizes how access to the flash memory module is affected by security and operating mode.

Table 23-51. Flash Memory Access Summary

Operating Mode	Chip Security State	
	Unsecure	Secure
NVM Normal	Full command set	
NVM Special	Full command set	Only the Erase All BlocksErase All Blocks Unsecure and Read 1s All Blocks commands.

23.4.11.2 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 affect after the next chip reset.

23.4.11.2.1 Unsecuring the Chip Using Backdoor Key Access

The chip can be unsecured by using the backdoor key access feature, which requires knowledge of the contents of the 8-byte backdoor key value stored in the Flash Configuration Field (see Flash Configuration Field Description). If the FSEC[KEYEN] bits are in the enabled state, the Verify Backdoor Access Key command (see Verify Backdoor Access Key Command) can be run; it allows the user to present prospective keys for comparison to the stored keys. If the keys match, the FSEC[SEC] bits are changed to unsecure the chip. The entire 8-byte key cannot be all 0s or all 1s; that is, 0000_0000_0000_0000h and FFFF_FFFF_FFFF_FFFFh are not accepted by the Verify Backdoor Access Key command as valid comparison values. While the Verify Backdoor Access Key command is active, program flash memory is not available for read access and returns invalid data.

The user code stored in the program flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN bits are in the enabled state, the chip can be unsecured by the following backdoor key access sequence:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Verify Backdoor Access Key Command

2. If the Verify Backdoor Access Key command is successful, the chip is unsecured and the FSEC[SEC] bits are forced to the unsecure state

An illegal key provided to the Verify Backdoor Access Key command prohibits further use of the Verify Backdoor Access Key command. A reset of the chip is the only method to re-enable the Verify Backdoor Access Key command when a comparison fails.

After the backdoor keys have been correctly matched, the chip is unsecured by changing the FSEC[SEC] bits. A successful execution of the Verify Backdoor Access Key command changes the security in the FSEC register only. It does not alter the security byte or the keys stored in the Flash Configuration Field (Flash Configuration Field Description). After the next reset of the chip, the security state of the flash memory module reverts back to the flash security byte in the Flash Configuration Field. The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the program flash protection registers.

If the backdoor keys successfully match, the unsecured chip has full control of the contents of the Flash Configuration Field. The chip may erase the sector containing the Flash Configuration Field and reprogram the flash security byte to the unsecure state and change the backdoor keys to any desired value.

23.4.12 Reset Sequence

On each system reset the flash memory module executes a sequence which establishes initial values for the flash block configuration parameters, FPROT, FOPT, and FSEC registers.

FSTAT[CCIF] is cleared throughout the reset sequence. The flash memory module holds off CPU access during the reset sequence. Flash reads are possible when the hold is removed. Completion of the reset sequence is marked by setting CCIF which enables flash user commands.

If a reset occurs while any flash command is in progress, that command is immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed. Commands and operations do not automatically resume after exiting reset.

Chapter 24
Analog-to-Digital Converter (ADC)

24.1 Introduction

NOTE

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

The 16-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

NOTE

For the chip specific modes of operation, see the power management information of the device.

24.1.1 Features

Following are the features of the ADC module.

- Linear successive approximation algorithm with up to 16-bit resolution
- Up to four pairs of differential and 24 single-ended external analog inputs
- Output modes:
  - differential 16-bit, 13-bit, 11-bit, and 9-bit modes
  - single-ended 16-bit, 12-bit, 10-bit, and 8-bit modes
- Output format in 2's complement 16-bit sign extended for differential modes
- Output in right-justified unsigned format for single-ended
- Single or continuous conversion, that is, automatic return to idle after single conversion

- Configurable sample time and conversion speed/power
- Conversion complete/hardware average complete flag and interrupt
- Input clock selectable from up to four sources
- Operation in low-power modes for lower noise
- Asynchronous clock source for lower noise operation with option to output the clock
- Selectable hardware conversion trigger with hardware channel select
- Automatic compare with interrupt for less-than, greater-than or equal-to, within range, or out-of-range, programmable value
- Temperature sensor
- Hardware average function
- Selectable voltage reference: external or alternate
- Self-Calibration mode

24.1.2 Block diagram

The following figure is the ADC module block diagram.

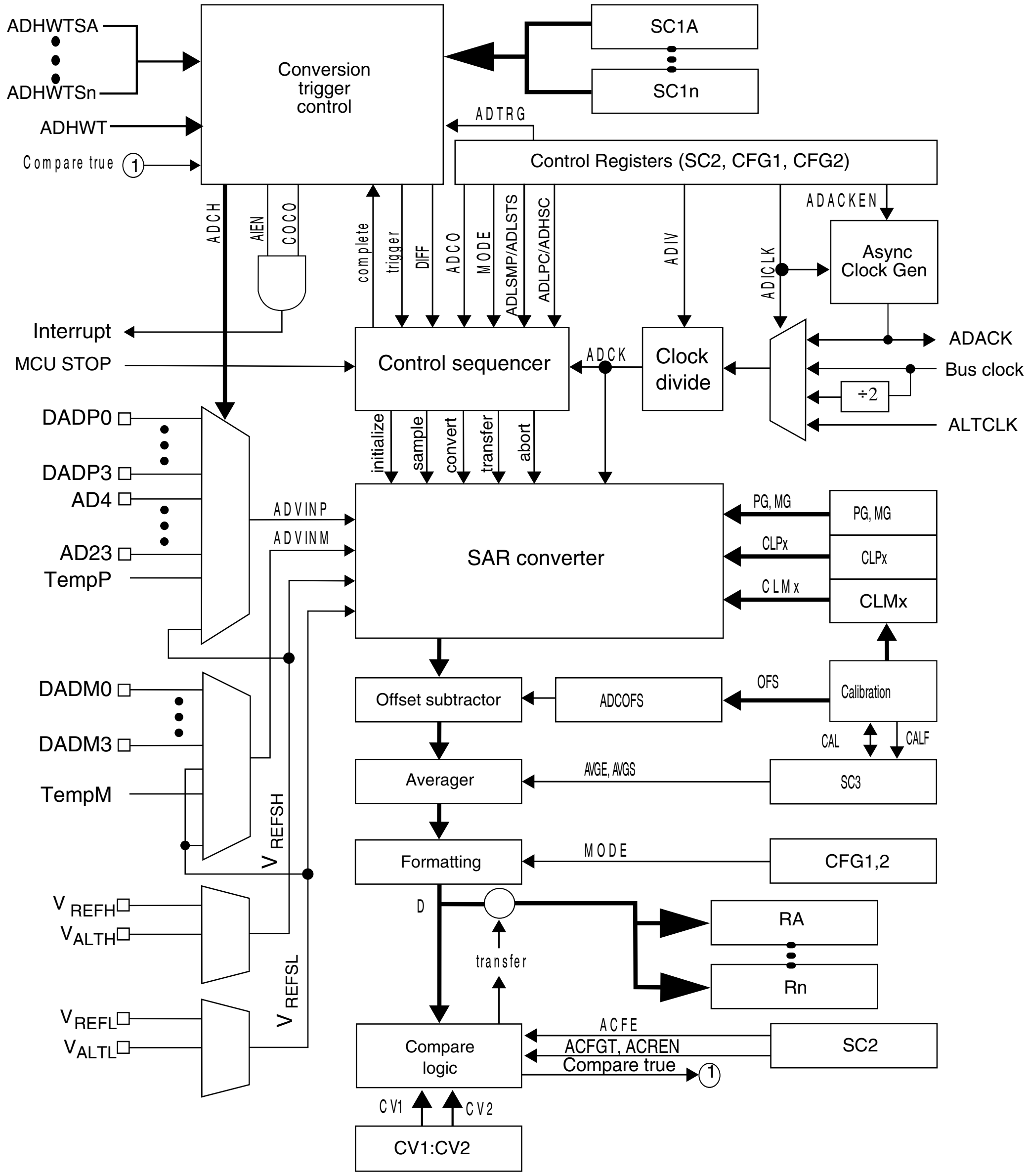


Figure 24-1. ADC block diagram

24.2 ADC signal descriptions

The ADC module supports up to 4 pairs of differential inputs and up to 24 single-ended inputs.

Each differential pair requires two inputs, DADPx and DADMx. The ADC also requires four supply/reference/ground connections.

NOTE

Refer to ADC configuration section in chip configuration chapter for the number of channels supported on this device.

Table 24-1. ADC signal descriptions

Signal	Description	I/O
DADP3–DADP0	Differential Analog Channel Inputs	I
DADM3–DADM0	Differential Analog Channel Inputs	I
AD*n*	Single-Ended Analog Channel Inputs	I
V_{REFSH}	Voltage Reference Select High	I
V_{REFSL}	Voltage Reference Select Low	I
V_{DDA}	Analog Power Supply	I
V_{SSA}	Analog Ground	I

24.2.1 Analog Power (V_{DDA})

The ADC analog portion uses V_{DDA} as its power connection. In some packages, V_{DDA} is connected internally to V_{DD}. If externally available, connect the V_{DDA} pin to the same voltage potential as V_{DD}. External filtering may be necessary to ensure clean V_{DDA} for good results.

24.2.2 Analog Ground (V_{SSA})

The ADC analog portion uses V_{SSA} as its ground connection. In some packages, V_{SSA} is connected internally to V_{SS}. If externally available, connect the V_{SSA} pin to the same voltage potential as V_{SS}.

24.2.3 Voltage Reference Select

V_{REFSH} and V_{REFSL} are the high and low reference voltages for the ADC module.

The ADC can be configured to accept one of two voltage reference pairs for V_{REFSH} and V_{REFSL}. Each pair contains a positive reference that must be between the minimum Ref Voltage High and V_{DDA}, and a ground reference that must be at the same potential as V_{SSA}. The two pairs are external (V_{REFH} and V_{REFL}) and alternate (V_{ALTH} and V_{ALTL}). These voltage references are selected using SC2[REFSEL]. The alternate V_{ALTH} and V_{ALTL} voltage reference pair may select additional external pins or internal sources depending on MCU configuration. See the chip configuration information on the Voltage References specific to this MCU.

In some packages, V_{REFH} is connected in the package to V_{DDA} and V_{REFL} to V_{SSA}. If externally available, the positive reference(s) may be connected to the same potential as V_{DDA} or may be driven by an external source to a level between the minimum Ref Voltage High and the V_{DDA} potential. V_{REFH} must never exceed V_{DDA}. Connect the ground references to the same voltage potential as V_{SSA}.

24.2.4 Analog Channel Inputs (ADx)

The ADC module supports up to 24 single-ended analog inputs. A single-ended input is selected for conversion through the SC1[ADCH] channel select bits when SC1n[DIFF] is low.

24.2.5 Differential Analog Channel Inputs (DADx)

The ADC module supports up to four differential analog channel inputs. Each differential analog input is a pair of external pins, DADPx and DADMx, referenced to each other to provide the most accurate analog to digital readings. A differential input is selected for conversion through SC1[ADCH] when SC1n[DIFF] is high. All DADPx inputs may be used as single-ended inputs if SC1n[DIFF] is low. In certain MCU configurations, some DADMx inputs may also be used as single-ended inputs if SC1n[DIFF] is low. See the chip configuration chapter for ADC connections specific to this MCU.

24.3 Memory map and register definitions

This section describes the ADC registers.

ADC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B000	ADC Status and Control Registers 1 (ADC0_SC1A)	32	R/W	0000_001Fh	24.3.1/424
4003_B004	ADC Status and Control Registers 1 (ADC0_SC1B)	32	R/W	0000_001Fh	24.3.1/424
4003_B008	ADC Configuration Register 1 (ADC0_CFG1)	32	R/W	0000_0000h	24.3.2/427
4003_B00C	ADC Configuration Register 2 (ADC0_CFG2)	32	R/W	0000_0000h	24.3.3/429
4003_B010	ADC Data Result Register (ADC0_RA)	32	R	0000_0000h	24.3.4/430
4003_B014	ADC Data Result Register (ADC0_RB)	32	R	0000_0000h	24.3.4/430
4003_B018	Compare Value Registers (ADC0_CV1)	32	R/W	0000_0000h	24.3.5/431
4003_B01C	Compare Value Registers (ADC0_CV2)	32	R/W	0000_0000h	24.3.5/431

Table continues on the next page...

ADC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B020	Status and Control Register 2 (ADC0_SC2)	32	R/W	0000_0000h	24.3.6/432
4003_B024	Status and Control Register 3 (ADC0_SC3)	32	R/W	0000_0000h	24.3.7/434
4003_B028	ADC Offset Correction Register (ADC0_OFS)	32	R/W	0000_0004h	24.3.8/436
4003_B02C	ADC Plus-Side Gain Register (ADC0_PG)	32	R/W	0000_8200h	24.3.9/436
4003_B030	ADC Minus-Side Gain Register (ADC0_MG)	32	R/W	0000_8200h	24.3.10/437
4003_B034	ADC Plus-Side General Calibration Value Register (ADC0_CLPD)	32	R/W	0000_000Ah	24.3.11/437
4003_B038	ADC Plus-Side General Calibration Value Register (ADC0_CLPS)	32	R/W	0000_0020h	24.3.12/438
4003_B03C	ADC Plus-Side General Calibration Value Register (ADC0_CLP4)	32	R/W	0000_0200h	24.3.13/438
4003_B040	ADC Plus-Side General Calibration Value Register (ADC0_CLP3)	32	R/W	0000_0100h	24.3.14/439
4003_B044	ADC Plus-Side General Calibration Value Register (ADC0_CLP2)	32	R/W	0000_0080h	24.3.15/439
4003_B048	ADC Plus-Side General Calibration Value Register (ADC0_CLP1)	32	R/W	0000_0040h	24.3.16/440
4003_B04C	ADC Plus-Side General Calibration Value Register (ADC0_CLP0)	32	R/W	0000_0020h	24.3.17/440
4003_B054	ADC Minus-Side General Calibration Value Register (ADC0_CLMD)	32	R/W	0000_000Ah	24.3.18/441
4003_B058	ADC Minus-Side General Calibration Value Register (ADC0_CLMS)	32	R/W	0000_0020h	24.3.19/441
4003_B05C	ADC Minus-Side General Calibration Value Register (ADC0_CLM4)	32	R/W	0000_0200h	24.3.20/442
4003_B060	ADC Minus-Side General Calibration Value Register (ADC0_CLM3)	32	R/W	0000_0100h	24.3.21/442
4003_B064	ADC Minus-Side General Calibration Value Register (ADC0_CLM2)	32	R/W	0000_0080h	24.3.22/443
4003_B068	ADC Minus-Side General Calibration Value Register (ADC0_CLM1)	32	R/W	0000_0040h	24.3.23/443
4003_B06C	ADC Minus-Side General Calibration Value Register (ADC0_CLM0)	32	R/W	0000_0020h	24.3.24/444

24.3.1 ADC Status and Control Registers 1 (ADCx_SC1*n*)

SC1A is used for both software and hardware trigger modes of operation.

To allow sequential conversions of the ADC to be triggered by internal peripherals, the ADC can have more than one status and control register: one for each conversion. The SC1B–SC1n registers indicate potentially multiple SC1 registers for use only in hardware trigger mode. See the chip configuration information about the number of SC1n registers specific to this device. The SC1n registers have identical fields, and are used in a "ping-pong" approach to control ADC operation.

At any one point in time, only one of the SC1n registers is actively controlling ADC conversions. Updating SC1A while SC1n is actively controlling a conversion is allowed, and vice-versa for any of the SC1n registers specific to this MCU.

Writing SC1A while SC1A is actively controlling a conversion aborts the current conversion. In Software Trigger mode, when SC2[ADTRG]=0, writes to SC1A subsequently initiate a new conversion, if SC1[ADCH] contains a value other than all 1s.

Writing any of the SC1n registers while that specific SC1n register is actively controlling a conversion aborts the current conversion. None of the SC1B-SC1n registers are used for software trigger operation and therefore writes to the SC1B–SC1n registers do not initiate a new conversion.

Address: 4003_B000h base + 0h offset + (4d × i), where i=0d to 1d

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

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

ADCx_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

Table continues on the next page...

ADCx_SC1n field descriptions (continued)

Field	Description
	This is a read-only field that is set each time a conversion is completed when the compare function is disabled, or SC2[ACFE]=0 and the hardware average function is disabled, or SC3[AVGE]=0. When the compare function is enabled, or SC2[ACFE]=1, COCO is set upon completion of a conversion only if the compare result is true. When the hardware average function is enabled, or SC3[AVGE]=1, COCO is set upon completion of the selected number of conversions (determined by AVGS). COCO in SC1A is also set at the completion of a calibration sequence. COCO is cleared when the respective SC1n register is written or when the respective Rn register is read. 0 Conversion is not completed. 1 Conversion is completed.
6 AIEN	Interrupt Enable Enables conversion complete interrupts. When COCO becomes set while the respective AIEN is high, an interrupt is asserted. 0 Conversion complete interrupt is disabled. 1 Conversion complete interrupt is enabled.
5 DIFF	Differential Mode Enable Configures the ADC to operate in differential mode. When enabled, this mode automatically selects from the differential channels, and changes the conversion algorithm and the number of cycles to complete a conversion. 0 Single-ended conversions and input channels are selected. 1 Differential conversions and input channels are selected.
4–0 ADCH	Input channel select Selects one of the input channels. The input channel decode depends on the value of DIFF. DAD0-DAD3 are associated with the input pin pairs DADPx and DADMx. **NOTE:** Some of the input channel options in the bitfield-setting descriptions might not be available for your device. For the actual ADC channel assignments for your device, see the Chip Configuration details. The successive approximation converter subsystem is turned off when the channel select bits are all set, that is, ADCH = 11111. This feature allows explicit disabling of the ADC and isolation of the input channel from all sources. Terminating continuous conversions this way prevents an additional single conversion from being performed. It is not necessary to set ADCH to all 1s to place the ADC in a low-power state when continuous conversions are not enabled because the module automatically enters a low-power state when a conversion completes. 00000 When DIFF=0, DADP0 is selected as input; when DIFF=1, DAD0 is selected as input. 00001 When DIFF=0, DADP1 is selected as input; when DIFF=1, DAD1 is selected as input. 00010 When DIFF=0, DADP2 is selected as input; when DIFF=1, DAD2 is selected as input. 00011 When DIFF=0, DADP3 is selected as input; when DIFF=1, DAD3 is selected as input. 00100 When DIFF=0, AD4 is selected as input; when DIFF=1, it is reserved. 00101 When DIFF=0, AD5 is selected as input; when DIFF=1, it is reserved. 00110 When DIFF=0, AD6 is selected as input; when DIFF=1, it is reserved. 00111 When DIFF=0, AD7 is selected as input; when DIFF=1, it is reserved. 01000 When DIFF=0, AD8 is selected as input; when DIFF=1, it is reserved. 01001 When DIFF=0, AD9 is selected as input; when DIFF=1, it is reserved. 01010 When DIFF=0, AD10 is selected as input; when DIFF=1, it is reserved. 01011 When DIFF=0, AD11 is selected as input; when DIFF=1, it is reserved.

Table continues on the next page...

ADCx_SC1n field descriptions (continued)

Field	Description
	01100 When DIFF=0, AD12 is selected as input; when DIFF=1, it is reserved. 01101 When DIFF=0, AD13 is selected as input; when DIFF=1, it is reserved. 01110 When DIFF=0, AD14 is selected as input; when DIFF=1, it is reserved. 01111 When DIFF=0, AD15 is selected as input; when DIFF=1, it is reserved. 10000 When DIFF=0, AD16 is selected as input; when DIFF=1, it is reserved. 10001 When DIFF=0, AD17 is selected as input; when DIFF=1, it is reserved. 10010 When DIFF=0, AD18 is selected as input; when DIFF=1, it is reserved. 10011 When DIFF=0, AD19 is selected as input; when DIFF=1, it is reserved. 10100 When DIFF=0, AD20 is selected as input; when DIFF=1, it is reserved. 10101 When DIFF=0, AD21 is selected as input; when DIFF=1, it is reserved. 10110 When DIFF=0, AD22 is selected as input; when DIFF=1, it is reserved. 10111 When DIFF=0, AD23 is selected as input; when DIFF=1, it is reserved. 11000 Reserved. 11001 Reserved. 11010 When DIFF=0, Temp Sensor (single-ended) is selected as input; when DIFF=1, Temp Sensor (differential) is selected as input. 11011 When DIFF=0,Bandgap (single-ended) is selected as input; when DIFF=1, Bandgap (differential) is selected as input. 11100 Reserved. 11101 When DIFF=0,V_{REFSH} is selected as input; when DIFF=1, -V_{REFSH} (differential) is selected as input. Voltage reference selected is determined by SC2[REFSEL]. 11110 When DIFF=0,V_{REFSL} is selected as input; when DIFF=1, it is reserved. Voltage reference selected is determined by SC2[REFSEL]. 11111 Module is disabled.

24.3.2 ADC Configuration Register 1 (ADCx_CFG1)

The configuration Register 1 (CFG1) selects the mode of operation, clock source, clock divide, and configuration for low power or long sample time.

Address: 4003_B000h base + 8h offset = 4003_B008h

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

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

ADCx_CFG1 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADLPC	Low-Power Configuration Controls the power configuration of the successive approximation converter. This optimizes power consumption when higher sample rates are not required. 0 Normal power configuration. 1 Low-power configuration. The power is reduced at the expense of maximum clock speed.
6–5 ADIV	Clock Divide Select Selects the divide ratio used by the ADC to generate the internal clock ADCK. 00 The divide ratio is 1 and the clock rate is input clock. 01 The divide ratio is 2 and the clock rate is (input clock)/2. 10 The divide ratio is 4 and the clock rate is (input clock)/4. 11 The divide ratio is 8 and the clock rate is (input clock)/8.
4 ADLSMP	Sample Time Configuration Selects between different sample times based on the conversion mode selected. This field adjusts the sample period to allow higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption if continuous conversions are enabled and high conversion rates are not required. When ADLSMP=1, the long sample time select bits, (ADLSTS[1:0]), can select the extent of the long sample time. 0 Short sample time. 1 Long sample time.
3–2 MODE	Conversion mode selection Selects the ADC resolution mode. 00 When DIFF=0:It is single-ended 8-bit conversion; when DIFF=1, it is differential 9-bit conversion with 2's complement output. 01 When DIFF=0:It is single-ended 12-bit conversion ; when DIFF=1, it is differential 13-bit conversion with 2's complement output. 10 When DIFF=0:It is single-ended 10-bit conversion. ; when DIFF=1, it is differential 11-bit conversion with 2's complement output 11 When DIFF=0:It is single-ended 16-bit conversion..; when DIFF=1, it is differential 16-bit conversion with 2's complement output
1–0 ADICLK	Input Clock Select Selects the input clock source to generate the internal clock, ADCK. Note that when the ADACK clock source is selected, it is not required to be active prior to conversion start. When it is selected and it is not active prior to a conversion start, when CFG2[ADACKEN]=0, the asynchronous clock is activated at the start of a conversion and deactivated when conversions are terminated. In this case, there is an associated clock startup delay each time the clock source is re-activated. 00 Bus clock 01 Bus clock divided by 2(BUSCLK/DIV2) 10 Alternate clock (ALTCLK) 11 Asynchronous clock (ADACK)

24.3.3 ADC Configuration Register 2 (ADCx_CFG2)

Configuration Register 2 (CFG2) selects the special high-speed configuration for very high speed conversions and selects the long sample time duration during long sample mode.

Address: 4003_B000h base + Ch offset = 4003_B00Ch

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

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

ADCx_CFG2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 MUXSEL	ADC Mux Select Changes the ADC mux setting to select between alternate sets of ADC channels. 0 ADxxa channels are selected. 1 ADxxb channels are selected.
3 ADACKEN	Asynchronous Clock Output Enable Enables the asynchronous clock source and the clock source output regardless of the conversion and status of CFG1[ADICLK]. Based on MCU configuration, the asynchronous clock may be used by other modules. See chip configuration information. Setting this field allows the clock to be used even while the ADC is idle or operating from a different clock source. Also, latency of initiating a single or first-continuous conversion with the asynchronous clock selected is reduced because the ADACK clock is already operational. 0 Asynchronous clock output disabled; Asynchronous clock is enabled only if selected by ADICLK and a conversion is active. 1 Asynchronous clock and clock output is enabled regardless of the state of the ADC.
2 ADHSC	High-Speed Configuration

Table continues on the next page...

ADCx_CFG2 field descriptions (continued)

Field	Description
	Configures the ADC for very high-speed operation. The conversion sequence is altered with 2 ADCK cycles added to the conversion time to allow higher speed conversion clocks. 0 Normal conversion sequence selected. 1 High-speed conversion sequence selected with 2 additional ADCK cycles to total conversion time.
1–0 ADLSTS	Long Sample Time Select Selects between the extended sample times when long sample time is selected, that is, when CFG1[ADLSMP]=1. This allows higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption when continuous conversions are enabled if high conversion rates are not required. 00 Default longest sample time; 20 extra ADCK cycles; 24 ADCK cycles total. 01 12 extra ADCK cycles; 16 ADCK cycles total sample time. 10 6 extra ADCK cycles; 10 ADCK cycles total sample time. 11 2 extra ADCK cycles; 6 ADCK cycles total sample time.

24.3.4 ADC Data Result Register (ADC*x*_R*n*)

The data result registers (Rn) contain the result of an ADC conversion of the channel selected by the corresponding status and channel control register (SC1A:SC1n). For every status and channel control register, there is a corresponding data result register.

Unused bits in R n are cleared in unsigned right-aligned modes and carry the sign bit (MSB) in sign-extended 2's complement modes. For example, when configured for 10-bit single-ended mode, D[15:10] are cleared. When configured for 11-bit differential mode, D[15:10] carry the sign bit, that is, bit 10 extended through bit 15.

The following table describes the behavior of the data result registers in the different modes of operation.

Table 24-43. Data result register description

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
16-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Signed 2's complement
16-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right justified
13-bit differential	S	S	S	S	D	D	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement
12-bit single-ended	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
11-bit differential	S	S	S	S	S	S	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement

Table continues on the next page...

Table 24-43. Data result register description (continued)

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
10-bit single-ended	0	0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
9-bit differential	S	S	S	S	S	S	S	S	D	D	D	D	D	D	D	D	Sign-extended 2's complement
8-bit single-ended	0	0	0	0	0	0	0	0	D	D	D	D	D	D	D	D	Unsigned right-justified

NOTE

S: Sign bit or sign bit extension;

D: Data, which is 2's complement data if indicated

Address: 4003_B000h base + 10h offset + (4d × i), where i=0d to 1d

Bit	31–16	15–0
R	0	D
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

ADCx_Rn field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 D	Data result

24.3.5 Compare Value Registers (ADCx_CVn)

The Compare Value Registers (CV1 and CV2) contain a compare value used to compare the conversion result when the compare function is enabled, that is, SC2[ACFE]=1. This register is formatted in the same way as the Rn registers in different modes of operation for both bit position definition and value format using unsigned or sign-extended 2's complement. Therefore, the compare function uses only the CVn fields that are related to the ADC mode of operation.

The compare value 2 register (CV2) is used only when the compare range function is enabled, that is, SC2[ACREN]=1.

Address: 4003_B000h base + 18h offset + (4d × i), where i=0d to 1d

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

ADCx_CV*n* field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 CV	Compare Value.

24.3.6 Status and Control Register 2 (ADCx_SC2)

The status and control register 2 (SC2) contains the conversion active, hardware/software trigger select, compare function, and voltage reference select of the ADC module.

Address: 4003_B000h base + 20h offset = 4003_B020h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADACT	ADTRG	ACFE	ACFGT	ACREN	DMAEN	REFSEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_SC2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADACT	Conversion Active Indicates that a conversion or hardware averaging is in progress. ADACT is set when a conversion is initiated and cleared when a conversion is completed or aborted. 0 Conversion not in progress. 1 Conversion in progress.
6 ADTRG	Conversion Trigger Select Selects the type of trigger used for initiating a conversion. Two types of trigger are selectable: • Software trigger: When software trigger is selected, a conversion is initiated following a write to SC1A. • Hardware trigger: When hardware trigger is selected, a conversion is initiated following the assertion of the ADHWT input after a pulse of the ADHWTSn input. 0 Software trigger selected. 1 Hardware trigger selected.
5 ACFE	Compare Function Enable Enables the compare function. 0 Compare function disabled. 1 Compare function enabled.
4 ACFGT	Compare Function Greater Than Enable Configures the compare function to check the conversion result relative to the CV1 and CV2 based upon the value of ACREN. ACFE must be set for ACFGT to have any effect. 0 Configures less than threshold, outside range not inclusive and inside range not inclusive; functionality based on the values placed in CV1 and CV2. 1 Configures greater than or equal to threshold, outside and inside ranges inclusive; functionality based on the values placed in CV1 and CV2.
3 ACREN	Compare Function Range Enable Configures the compare function to check if the conversion result of the input being monitored is either between or outside the range formed by CV1 and CV2 determined by the value of ACFGT. ACFE must be set for ACFGT to have any effect. 0 Range function disabled. Only CV1 is compared. 1 Range function enabled. Both CV1 and CV2 are compared.
2 DMAEN	DMA Enable 0 DMA is disabled. 1 DMA is enabled and will assert the ADC DMA request during an ADC conversion complete event noted when any of the SC1n[COCO] flags is asserted.
1–0 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}

Table continues on the next page...

ADCx_SC2 field descriptions (continued)

Field	Description
	01 Alternate reference pair, that is, V_{ALTH} and V_{ALTL} . This pair may be additional external pins or internal sources depending on the MCU configuration. See the chip configuration information for details specific to this MCU 10 Reserved 11 Reserved

24.3.7 Status and Control Register 3 (ADCx_SC3)

The Status and Control Register 3 (SC3) controls the calibration, continuous convert, and hardware averaging functions of the ADC module.

Address: 4003_B000h base + 24h offset = 4003_B024h

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

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

ADCx_SC3 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ADCx_SC3 field descriptions (continued)

Field	Description
7 CAL	Calibration Begins the calibration sequence when set. This field stays set while the calibration is in progress and is cleared when the calibration sequence is completed. CALF must be checked to determine the result of the calibration sequence. Once started, the calibration routine cannot be interrupted by writes to the ADC registers or the results will be invalid and CALF will set. Setting CAL will abort any current conversion.
6 CALF	Calibration Failed Flag Displays the result of the calibration sequence. The calibration sequence will fail if SC2[ADTRG] = 1, any ADC register is written, or any stop mode is entered before the calibration sequence completes. Writing 1 to CALF clears it. 0 Calibration completed normally. 1 Calibration failed. ADC accuracy specifications are not guaranteed.
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.
1–0 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.

24.3.8 ADC Offset Correction Register (ADCx_OFS)

The ADC Offset Correction Register (OFS) contains the user-selected or calibration-generated offset error correction value. This register is a 2's complement, left-justified, 16-bit value . The value in OFS is subtracted from the conversion and the result is transferred into the result registers, Rn. If the result is greater than the maximum or less than the minimum result value, it is forced to the appropriate limit for the current mode of operation.

Address: 4003_B000h base + 28h offset = 4003_B028h

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

ADCx_OFS field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 OFS	Offset Error Correction Value

24.3.9 ADC Plus-Side Gain Register (ADCx_PG)

The Plus-Side Gain Register (PG) contains the gain error correction for the plus-side input in differential mode or the overall conversion in single-ended mode. PG, a 16-bit real number in binary format, is the gain adjustment factor, with the radix point fixed between ADPG15 and ADPG14. This register must be written by the user with the value described in the calibration procedure. Otherwise, the gain error specifications may not be met.

Address: 4003_B000h base + 2Ch offset = 4003_B02Ch

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

ADCx_PG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 PG	Plus-Side Gain

24.3.10 ADC Minus-Side Gain Register (ADCx_MG)

The Minus-Side Gain Register (MG) contains the gain error correction for the minus-side input in differential mode. This register is ignored in single-ended mode. MG, a 16-bit real number in binary format, is the gain adjustment factor, with the radix point fixed between ADMG15 and ADMG14. This register must be written by the user with the value described in the calibration procedure. Otherwise, the gain error specifications may not be met.

Address: 4003_B000h base + 30h offset = 4003_B030h

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

ADCx_MG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 MG	Minus-Side Gain

24.3.11 ADC Plus-Side General Calibration Value Register (ADCx_CLPD)

The Plus-Side General Calibration Value Registers (CLPx) contain calibration information that is generated by the calibration function. These registers contain seven calibration values of varying widths: CLP0[5:0], CLP1[6:0], CLP2[7:0], CLP3[8:0], CLP4[9:0], CLPS[5:0], and CLPD[5:0]. CLPx are automatically set when the self-calibration sequence is done, that is, CAL is cleared. If these registers are written by the user after calibration, the linearity error specifications may not be met.

Address: 4003_B000h base + 34h offset = 4003_B034h

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

ADCx_CLPD field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 CLPD	Calibration Value

24.3.12 ADC Plus-Side General Calibration Value Register (ADCx_CLPS)

For more information, see CLPD register description.

Address: 4003_B000h base + 38h offset = 4003_B038h

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

ADCx_CLPS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 CLPS	Calibration Value

24.3.13 ADC Plus-Side General Calibration Value Register (ADCx_CLP4)

For more information, see CLPD register description.

Address: 4003_B000h base + 3Ch offset = 4003_B03Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						CLP4									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_CLP4 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–0 CLP4	Calibration Value

24.3.14 ADC Plus-Side General Calibration Value Register (ADCx_CLP3)

For more information, see CLPD register description.

Address: 4003_B000h base + 40h offset = 4003_B040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																							CLP3								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

ADCx_CLP3 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–0 CLP3	Calibration Value

24.3.15 ADC Plus-Side General Calibration Value Register (ADCx_CLP2)

For more information, see CLPD register description.

Address: 4003_B000h base + 44h offset = 4003_B044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								CLP2							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

ADCx_CLP2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 CLP2	Calibration Value

24.3.16 ADC Plus-Side General Calibration Value Register (ADCx_CLP1)

For more information, see CLPD register description.

Address: 4003_B000h base + 48h offset = 4003_B048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									CLP1						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

ADCx_CLP1 field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 CLP1	Calibration Value

24.3.17 ADC Plus-Side General Calibration Value Register (ADCx_CLP0)

For more information, see CLPD register description.

Address: 4003_B000h base + 4Ch offset = 4003_B04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLP0					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLP0 field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 CLP0	Calibration Value

24.3.18 ADC Minus-Side General Calibration Value Register (ADCx_CLMD)

The Minus-Side General Calibration Value (CLMx) registers contain calibration information that is generated by the calibration function. These registers contain seven calibration values of varying widths: CLM0[5:0], CLM1[6:0], CLM2[7:0], CLM3[8:0], CLM4[9:0], CLMS[5:0], and CLMD[5:0]. CLMx are automatically set when the self-calibration sequence is done, that is, CAL is cleared. If these registers are written by the user after calibration, the linearity error specifications may not be met.

Address: 4003_B000h base + 54h offset = 4003_B054h

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

ADCx_CLMD field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 CLMD	Calibration Value

24.3.19 ADC Minus-Side General Calibration Value Register (ADCx_CLMS)

For more information, see CLMD register description.

Address: 4003_B000h base + 58h offset = 4003_B058h

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

ADCx_CLMS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 CLMS	Calibration Value

24.3.20 ADC Minus-Side General Calibration Value Register (ADCx_CLM4)

For more information, see CLMD register description.

Address: 4003_B000h base + 5Ch offset = 4003_B05Ch

Bit	31–10	9–0
R	0	CLM4
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0 0

ADCx_CLM4 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–0 CLM4	Calibration Value

24.3.21 ADC Minus-Side General Calibration Value Register (ADCx_CLM3)

For more information, see CLMD register description.

Address: 4003_B000h base + 60h offset = 4003_B060h

Bit	31–9	8–0
R	0	CLM3
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0

ADCx_CLM3 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–0 CLM3	Calibration Value

24.3.22 ADC Minus-Side General Calibration Value Register (ADCx_CLM2)

For more information, see CLMD register description.

Address: 4003_B000h base + 64h offset = 4003_B064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								CLM2							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

ADCx_CLM2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 CLM2	Calibration Value

24.3.23 ADC Minus-Side General Calibration Value Register (ADCx_CLM1)

For more information, see CLMD register description.

Address: 4003_B000h base + 68h offset = 4003_B068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									CLM1						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

ADCx_CLM1 field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 CLM1	Calibration Value

24.3.24 ADC Minus-Side General Calibration Value Register (ADCx_CLM0)

For more information, see CLMD register description.

Address: 4003_B000h base + 6Ch offset = 4003_B06Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLM0					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLM0 field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 CLM0	Calibration Value

24.4 Functional description

The ADC module is disabled during reset, in Low-Power Stop mode, or when SC1n[ADCH] are all high; see the power management information for details. The module is idle when a conversion has completed and another conversion has not been initiated. When it is idle and the asynchronous clock output enable is disabled, or CFG2[ADACKEN]= 0, the module is in its lowest power state. The ADC can perform an analog-to-digital conversion on any of the software selectable channels. All modes perform conversion by a successive approximation algorithm.

To meet accuracy specifications, the ADC module must be calibrated using the on-chip calibration function.

See Calibration function for details on how to perform calibration.

When the conversion is completed, the result is placed in the Rn data registers. The respective SC1n[COCO] is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled, or, when SC1n[AIEN]=1.

The ADC module has the capability of automatically comparing the result of a conversion with the contents of the CV1 and CV2 registers. The compare function is enabled by setting SC2[ACFE] and operates in any of the conversion modes and configurations.

The ADC module has the capability of automatically averaging the result of multiple conversions. The hardware average function is enabled by setting SC3[AVGE] and operates in any of the conversion modes and configurations.

NOTE

For the chip specific modes of operation, see the power management information of this MCU.

24.4.1 Clock select and divide control

One of four clock sources can be selected as the clock source for the ADC module.

This clock source is then divided by a configurable value to generate the input clock ADCK, to the module. The clock is selected from one of the following sources by means of CFG1[ADICLK].

- Bus clock. This is the default selection following reset.
- Bus clock divided by two. For higher bus clock rates, this allows a maximum divide-by-16 of the bus clock using CFG1[ADIV].
- ALTCLK: As defined for this MCU. See the chip configuration information. Conversions are possible using ALTCLK as the input clock source while the MCU is in Normal Stop mode.
- Asynchronous clock (ADACK): This clock is generated from a clock source within the ADC module. When the ADACK clock source is selected, it is not required to be active prior to conversion start. When it is selected and it is not active prior to a conversion start CFG2[ADACKEN]=0, ADACK is activated at the start of a conversion and deactivated when conversions are terminated. In this case, there is an associated clock startup delay each time the clock source is re-activated. To avoid the conversion time variability and latency associated with the ADACK clock startup, set CFG2[ADACKEN]=1 and wait the worst-case startup time of 5 µs prior to initiating any conversions using the ADACK clock source. Conversions are possible using ADACK as the input clock source while the MCU is in Normal Stop mode. See Power Control for more information.

Whichever clock is selected, its frequency must fall within the specified frequency range for ADCK. If the available clocks are too slow, the ADC may not perform according to specifications. If the available clocks are too fast, the clock must be divided to the appropriate frequency. This divider is specified by CFG1[ADIV] and can be divide-by 1, 2, 4, or 8.

24.4.2 Voltage reference selection

The ADC can be configured to accept one of the two voltage reference pairs as the reference voltage (V_{REFSH} and V_{REFSL}) used for conversions.

Each pair contains a positive reference that must be between the minimum Ref Voltage High and V_{DDA}, and a ground reference that must be at the same potential as V_{SSA}. The two pairs are external (V_{REFH} and V_{REFL}) and alternate (V_{ALTH} and V_{ALTL}). These voltage references are selected using SC2[REFSEL]. The alternate (V_{ALTH} and V_{ALTL}) voltage reference pair may select additional external pins or internal sources depending on MCU configuration. See the chip configuration information on the voltage references specific to this MCU.

24.4.3 Hardware trigger and channel selects

The ADC module has a selectable asynchronous hardware conversion trigger, ADHWT, that is enabled when SC2[ADTRG] is set and a hardware trigger select event, ADHWTSn, has occurred.

This source is not available on all MCUs. See the Chip Configuration chapter for information on the ADHWT source and the ADHWTSn configurations specific to this MCU.

When an ADHWT source is available and hardware trigger is enabled, that is SC2[ADTRG]=1, a conversion is initiated on the rising-edge of ADHWT after a hardware trigger select event, that is, ADHWTSn, has occurred. If a conversion is in progress when a rising-edge of a trigger occurs, the rising-edge is ignored. In continuous convert configuration, only the initial rising-edge to launch continuous conversions is observed, and until conversion is aborted, the ADC continues to do conversions on the same SCn register that initiated the conversion. The hardware trigger function operates in conjunction with any of the conversion modes and configurations.

The hardware trigger select event, ADHWTSn, must be set prior to the receipt of the ADHWT signal. If these conditions are not met, the converter may ignore the trigger or use the incorrect configuration. If a hardware trigger select event is asserted during a conversion, it must stay asserted until the end of current conversion and remain set until the receipt of the ADHWT signal to trigger a new conversion. The channel and status fields selected for the conversion depend on the active trigger select signal:

- ADHWTSA active selects SC1A.
- ADHWTSn active selects SC1n.

Note

Asserting more than one hardware trigger select signal (ADHWTSn) at the same time results in unknown results. To avoid this, select only one hardware trigger select signal (ADHWTSn) prior to the next intended conversion.

When the conversion is completed, the result is placed in the Rn registers associated with the ADHWTSn received. For example:

- ADHWTSA active selects RA register
- ADHWTSn active selects Rn register

The conversion complete flag associated with the ADHWTSn received, that is, SC1n[COCO], is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled, that is, SC1[AIEN]=1.

24.4.4 Conversion control

Conversions can be performed as determined by CFG1[MODE] and SC1n[DIFF] as shown in the description of CFG1[MODE].

Conversions can be initiated by a software or hardware trigger.

In addition, the ADC module can be configured for:

- Low-power operation
- Long sample time
- Continuous conversion
- Hardware average
- Automatic compare of the conversion result to a software determined compare value

24.4.4.1 Initiating conversions

A conversion is initiated:

- Following a write to SC1A, with SC1n[ADCH] not all 1's, if software triggered operation is selected, that is, when SC2[ADTRG]=0.
- Following a hardware trigger, or ADHWT event, if hardware triggered operation is selected, that is, SC2[ADTRG]=1, and a hardware trigger select event, ADHWTSn, has occurred. The channel and status fields selected depend on the active trigger select signal:
    - ADHWTSA active selects SC1A.

- ADHWTSn active selects SC1n.
- if neither is active, the off condition is selected

Note

Selecting more than one ADHWTSn prior to a conversion completion will result in unknown results. To avoid this, select only one ADHWTSn prior to a conversion completion.

- Following the transfer of the result to the data registers when continuous conversion is enabled, that is, when SC3[ADCO] = 1.

If continuous conversions are enabled, a new conversion is automatically initiated after the completion of the current conversion. In software triggered operation, that is, when SC2[ADTRG] = 0, continuous conversions begin after SC1A is written and continue until aborted. In hardware triggered operation, that is, when SC2[ADTRG] = 1 and one ADHWTSn event has occurred, continuous conversions begin after a hardware trigger event and continue until aborted.

If hardware averaging is enabled, a new conversion is automatically initiated after the completion of the current conversion until the correct number of conversions are completed. In software triggered operation, conversions begin after SC1A is written. In hardware triggered operation, conversions begin after a hardware trigger. If continuous conversions are also enabled, a new set of conversions to be averaged are initiated following the last of the selected number of conversions.

24.4.4.2 Completing conversions

A conversion is completed when the result of the conversion is transferred into the data result registers, Rn. If the compare functions are disabled, this is indicated by setting of SC1n[COCO]. If hardware averaging is enabled, the respective SC1n[COCO] sets only if the last of the selected number of conversions is completed. If the compare function is enabled, the respective SC1n[COCO] sets and conversion result data is transferred only if the compare condition is true. If both hardware averaging and compare functions are enabled, then the respective SC1n[COCO] sets only if the last of the selected number of conversions is completed and the compare condition is true. An interrupt is generated if the respective SC1n[AIEN] is high at the time that the respective SC1n[COCO] is set.

24.4.4.3 Aborting conversions

Any conversion in progress is aborted when:

- Writing to SC1A while it is actively controlling a conversion, aborts the current conversion. In Software Trigger mode, when SC2[ADTRG]=0, a write to SC1A initiates a new conversion if SC1A[ADCH] is equal to a value other than all 1s. Writing to any of the SC1B–SC1n registers while that specific SC1B–SC1n register is actively controlling a conversion aborts the current conversion. The SC1(B-n) registers are not used for software trigger operation and therefore writes to the SC1(B-n) registers do not initiate a new conversion.
- A write to any ADC register besides the SC1A-SC1n registers occurs. This indicates that a change in mode of operation has occurred and the current conversion is therefore invalid.
- The MCU is reset or enters Low-Power Stop modes.
- The MCU enters Normal Stop mode with ADACK or Alternate Clock Sources not enabled.

When a conversion is aborted, the contents of the data registers, Rn, are not altered. The data registers continue to be the values transferred after the completion of the last successful conversion. If the conversion was aborted by a reset or Low-Power Stop modes, RA and Rn return to their reset states.

24.4.4.4 Power control

The ADC module remains in its idle state until a conversion is initiated. If ADACK is selected as the conversion clock source, but the asynchronous clock output is disabled, that is CFG2[ADACKEN]=0, the ADACK clock generator also remains in its idle state (disabled) until a conversion is initiated. If the asynchronous clock output is enabled, that is, CFG2[ADACKEN]=1, it remains active regardless of the state of the ADC or the MCU power mode.

Power consumption when the ADC is active can be reduced by setting CFG1[ADLPC]. This results in a lower maximum value for f_{ADCK}.

24.4.4.5 Sample time and total conversion time

For short sample, that is, when CFG1[ADLSMP]=0, there is a 2-cycle adder for first conversion over the base sample time of four ADCK cycles. For high-speed conversions, that is, when CFG2[ADHSC]=1, there is an additional 2-cycle adder on any conversion. The table below summarizes sample times for the possible ADC configurations.

ADC configuration			Sample time (ADCK cycles)	
CFG1[ADLSMP]	CFG2[ADLSTS]	CFG2[ADHSC]	First or Single	Subsequent
0	X	0	6	4
1	00	0	24	
1	01	0	16	
1	10	0	10	
1	11	0	6	
0	X	1	8	6
1	00	1	26	
1	01	1	18	
1	10	1	12	
1	11	1	8	

The total conversion time depends upon:

- The sample time as determined by CFG1[ADLSMP] and CFG2[ADLSTS]
- The MCU bus frequency
- The conversion mode, as determined by CFG1[MODE] and SC1n[DIFF]
- The high-speed configuration, that is, CFG2[ADHSC]
- The frequency of the conversion clock, that is, f_{ADCK}.

CFG2[ADHSC] is used to configure a higher clock input frequency. This will allow faster overall conversion times. To meet internal ADC timing requirements, CFG2[ADHSC] adds additional ADCK cycles. Conversions with CFG2[ADHSC]=1 take two more ADCK cycles. CFG2[ADHSC] must be used when the ADCLK exceeds the limit for CFG2[ADHSC]=0.

After the module becomes active, sampling of the input begins.

1. CFG1[ADLSMP] and CFG2[ADLSTS] select between sample times based on the conversion mode that is selected.
2. When sampling is completed, the converter is isolated from the input channel and a successive approximation algorithm is applied to determine the digital value of the analog signal.
3. The result of the conversion is transferred to Rn upon completion of the conversion algorithm.

If the bus frequency is less than f_{ADCK}, precise sample time for continuous conversions cannot be guaranteed when short sample is enabled, that is, when CFG1[ADLSMP]=0.

The maximum total conversion time is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by CFG1[ADICLK], and the divide ratio is specified by CFG1[ADIV].

The maximum total conversion time for all configurations is summarized in the equation below. See the following tables for the variables referenced in the equation.

$$\mathrm{ConversionTime} = \mathrm{SFCAdder} + \mathrm{AverageNum} \times (\mathrm{BCT} + \mathrm{LSTAdder} + \mathrm{HSCAdder})$$

Figure 24-62. Conversion time equation

Table 24-70. Single or first continuous time adder (SFCAdder)

CFG1[ADLSMP]	CFG2[ADACKEN]	CFG1[ADICLK]	Single or first continuous time adder (SFCAdder)
1	x	0x, 10	3 ADCK cycles + 5 bus clock cycles
1	1	11	3 ADCK cycles + 5 bus clock cycles[1]
1	0	11	5 μs + 3 ADCK cycles + 5 bus clock cycles
0	x	0x, 10	5 ADCK cycles + 5 bus clock cycles
0	1	11	5 ADCK cycles + 5 bus clock cycles[1]
0	0	11	5 μs + 5 ADCK cycles + 5 bus clock cycles

1. To achieve this time, CFG2[ADACKEN] must be 1 for at least 5 μs prior to the conversion is initiated.

Table 24-71. Average number factor (AverageNum)

SC3[AVGE]	SC3[AVGS]	Average number factor (AverageNum)
0	xx	1
1	00	4
1	01	8
1	10	16
1	11	32

Table 24-72. Base conversion time (BCT)

Mode	Base conversion time (BCT)
8b single-ended	17 ADCK cycles
9b differential	27 ADCK cycles
10b single-ended	20 ADCK cycles
11b differential	30 ADCK cycles
12b single-ended	20 ADCK cycles
13b differential	30 ADCK cycles

Table continues on the next page...

Table 24-72. Base conversion time (BCT) (continued)

Mode	Base conversion time (BCT)
16b single-ended	25 ADCK cycles
16b differential	34 ADCK cycles

Table 24-73. Long sample time adder (LSTAdder)

CFG1[ADLSMP]	CFG2[ADLSTS]	Long sample time adder (LSTAdder)
0	xx	0 ADCK cycles
1	00	20 ADCK cycles
1	01	12 ADCK cycles
1	10	6 ADCK cycles
1	11	2 ADCK cycles

Table 24-74. High-speed conversion time adder (HSCAdder)

CFG2[ADHSC]	High-speed conversion time adder (HSCAdder)
0	0 ADCK cycles
1	2 ADCK cycles

Note

The ADCK frequency must be between f_{ADCK} minimum and f_{ADCK} maximum to meet ADC specifications.

24.4.4.6 Conversion time examples

The following examples use the Figure 24-62, and the information provided in Table 24-70 through Table 24-74.

24.4.4.6.1 Typical conversion time configuration

A typical configuration for ADC conversion is:

- 10-bit mode, with the bus clock selected as the input clock source
- The input clock divide-by-1 ratio selected
- Bus frequency of 8 MHz
- Long sample time disabled
- High-speed conversion disabled

The conversion time for a single conversion is calculated by using the Figure 24-62, and the information provided in Table 24-70 through Table 24-74. The table below lists the variables of Figure 24-62.

Table 24-75. Typical conversion time

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	20 ADCK cycles
LSTAdder	0
HSCAdder	0

The resulting conversion time is generated using the parameters listed in the preceding table. Therefore, for a bus clock and an ADCK frequency equal to 8 MHz, the resulting conversion time is 3.75 µs.

24.4.4.6.2 Long conversion time configuration

A configuration for long ADC conversion is:

- 16-bit differential mode with the bus clock selected as the input clock source
- The input clock divide-by-8 ratio selected
- Bus frequency of 8 MHz
- Long sample time enabled
- Configured for longest adder
- High-speed conversion disabled
- Average enabled for 32 conversions

The conversion time for this conversion is calculated by using the Figure 24-62, and the information provided in Table 24-70 through Table 24-74. The following table lists the variables of the Figure 24-62.

Table 24-76. Typical conversion time

Variable	Time
SFCAdder	3 ADCK cycles + 5 bus clock cycles
AverageNum	32
BCT	34 ADCK cycles
LSTAdder	20 ADCK cycles
HSCAdder	0

The resulting conversion time is generated using the parameters listed in the preceding table. Therefore, for bus clock equal to 8 MHz and ADCK equal to 1 MHz, the resulting conversion time is 57.625 µs, that is, AverageNum. This results in a total conversion time of 1.844 ms.

24.4.4.6.3 Short conversion time configuration

A configuration for short ADC conversion is:

- 8-bit Single-Ended mode with the bus clock selected as the input clock source
- The input clock divide-by-1 ratio selected
- Bus frequency of 20 MHz
- Long sample time disabled
- High-speed conversion enabled

The conversion time for this conversion is calculated by using the Figure 24-62, and the information provided in Table 24-70 through Table 24-74. The table below lists the variables of Figure 24-62.

Table 24-77. Typical conversion time

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	17 ADCK cycles
LSTAdder	0 ADCK cycles
HSCAdder	2

The resulting conversion time is generated using the parameters listed in in the preceding table. Therefore, for bus clock and ADCK frequency equal to 20 MHz, the resulting conversion time is 1.45 µs.

24.4.4.7 Hardware average function

The hardware average function can be enabled by setting SC3[AVGE]=1 to perform a hardware average of multiple conversions. The number of conversions is determined by the AVGS[1:0] bits, which can select 4, 8, 16, or 32 conversions to be averaged. While the hardware average function is in progress, SC2[ADACT] will be set.

After the selected input is sampled and converted, the result is placed in an accumulator from which an average is calculated once the selected number of conversions have been completed. When hardware averaging is selected, the completion of a single conversion will not set SC1n[COCO].

If the compare function is either disabled or evaluates true, after the selected number of conversions are completed, the average conversion result is transferred into the data result registers, Rn, and SC1n[COCO] is set. An ADC interrupt is generated upon the setting of SC1n[COCO] if the respective ADC interrupt is enabled, that is, SC1n[AIEN]=1.

Note

The hardware average function can perform conversions on a channel while the MCU is in Wait or Normal Stop modes. The ADC interrupt wakes the MCU when the hardware average is completed if SC1n[AIEN] is set.

24.4.5 Automatic compare function

The compare function can be configured to check whether the result is less than or greater-than-or-equal-to a single compare value, or, if the result falls within or outside a range determined by two compare values.

The compare mode is determined by SC2[ACFGT], SC2[ACREN], and the values in the compare value registers, CV1 and CV2. After the input is sampled and converted, the compare values in CV1 and CV2 are used as described in the following table. There are six Compare modes as shown in the following table.

Table 24-78. Compare modes

SC2[ACFGT]	SC2[ACREN]	ADCCV1 relative to ADCCV2	Function	Compare mode description
0	0	—	Less than threshold	Compare true if the result is less than the CV1 registers.
1	0	—	Greater than or equal to threshold	Compare true if the result is greater than or equal to CV1 registers.
0	1	Less than or equal	Outside range, not inclusive	Compare true if the result is less than CV1 **Or** the result is greater than CV2.
0	1	Greater than	Inside range, not inclusive	Compare true if the result is less than CV1 **And** the result is greater than CV2.
1	1	Less than or equal	Inside range, inclusive	Compare true if the result is greater than or equal to CV1 **And** the result is less than or equal to CV2.
1	1	Greater than	Outside range, inclusive	Compare true if the result is greater than or equal to CV1 **Or** the result is less than or equal to CV2.

With SC2[ACREN] =1, and if the value of CV1 is less than or equal to the value of CV2, then setting SC2[ACFGT] will select a trigger-if-inside-compare-range inclusive-of-endpoints function. Clearing SC2[ACFGT] will select a trigger-if-outside-compare-range, not-inclusive-of-endpoints function.

If CV1 is greater than CV2, setting SC2[ACFGT] will select a trigger-if-outside-compare-range, inclusive-of-endpoints function. Clearing SC2[ACFGT] will select a trigger-if-inside-compare-range, not-inclusive-of-endpoints function.

If the condition selected evaluates true, SC1n[COCO] is set.

Upon completion of a conversion while the compare function is enabled, if the compare condition is not true, SC1n[COCO] is not set and the conversion result data will not be transferred to the result register, Rn. If the hardware averaging function is enabled, the compare function compares the averaged result to the compare values. The same compare function definitions apply. An ADC interrupt is generated when SC1n[COCO] is set and the respective ADC interrupt is enabled, that is, SC1n[AIEN]=1.

Note

The compare function can monitor the voltage on a channel while the MCU is in Wait or Normal Stop modes. The ADC interrupt wakes the MCU when the compare condition is met.

24.4.6 Calibration function

>The ADC contains a self-calibration function that is required to achieve the specified accuracy.

Calibration must be run, or valid calibration values written, after any reset and before a conversion is initiated. The calibration function sets the offset calibration value, the minus-side calibration values, and the plus-side calibration values. The offset calibration value is automatically stored in the ADC offset correction register (OFS), and the plus-side and minus-side calibration values are automatically stored in the ADC plus-side and minus-side calibration registers, CLPx and CLMx. The user must configure the ADC correctly prior to calibration, and must generate the plus-side and minus-side gain calibration results and store them in the ADC plus-side gain register (PG) after the calibration function completes.

Prior to calibration, the user must configure the ADC's clock source and frequency, low power configuration, voltage reference selection, sample time, and high speed configuration according to the application's clock source availability and needs. If the application uses the ADC in a wide variety of configurations, the configuration for which the highest accuracy is required should be selected, or multiple calibrations can be done for the different configurations. For best calibration results:

- Set hardware averaging to maximum, that is, SC3[AVGE]=1 and SC3[AVGS]=11 for an average of 32
- Set ADC clock frequency f_{ADCK} less than or equal to 4 MHz
- $V_{REFH}=V_{DDA}$
- Calibrate at nominal voltage and temperature

The input channel, conversion mode continuous function, compare function, resolution mode, and differential/single-ended mode are all ignored during the calibration function.

To initiate calibration, the user sets SC3[CAL] and the calibration will automatically begin if the SC2[ADTRG] is 0. If SC2[ADTRG] is 1, SC3[CAL] will not get set and SC3[CALF] will be set. While calibration is active, no ADC register can be written and no stop mode may be entered, or the calibration routine will be aborted causing SC3[CAL] to clear and SC3[CALF] to set. At the end of a calibration sequence, SC1n[COCO] will be set. SC1n[AIEN] can be used to allow an interrupt to occur at the end of a calibration sequence. At the end of the calibration routine, if SC3[CALF] is not set, the automatic calibration routine is completed successfully.

To complete calibration, the user must generate the gain calibration values using the following procedure:

1. Initialize or clear a 16-bit variable in RAM.
2. Add the plus-side calibration results CLP0, CLP1, CLP2, CLP3, CLP4, and CLPS to the variable.
3. Divide the variable by two.
4. Set the MSB of the variable.
5. The previous two steps can be achieved by setting the carry bit, rotating to the right through the carry bit on the high byte and again on the low byte.
6. Store the value in the plus-side gain calibration register PG.
7. Repeat the procedure for the minus-side gain calibration value.

When calibration is complete, the user may reconfigure and use the ADC as desired. A second calibration may also be performed, if desired, by clearing and again setting SC3[CAL].

Overall, the calibration routine may take as many as 14k ADCK cycles and 100 bus cycles, depending on the results and the clock source chosen. For an 8 MHz clock source, this length amounts to about 1.7 ms. To reduce this latency, the calibration values, which are offset, plus-side and minus-side gain, and plus-side and minus-side calibration values, may be stored in flash memory after an initial calibration and recovered prior to the first ADC conversion. This method can reduce the calibration latency to 20 register store operations on all subsequent power, reset, or Low-Power Stop mode recoveries.

Further information on the calibration procedure can be found in the Calibration section of AN3949: ADC16 Calibration Procedure and Programmable Delay Block Synchronization.

24.4.7 User-defined offset function

OFS contains the user-selected or calibration-generated offset error correction value.

This register is a 2's complement, left-justified. The value in OFS is subtracted from the conversion and the result is transferred into the result registers, Rn. If the result is greater than the maximum or less than the minimum result value, it is forced to the appropriate limit for the current mode of operation.

The formatting of the OFS is different from the data result register, Rn, to preserve the resolution of the calibration value regardless of the conversion mode selected. Lower order bits are ignored in lower resolution modes. For example, in 8-bit single-ended mode, OFS[14:7] are subtracted from D[7:0]; OFS[15] indicates the sign (negative numbers are effectively added to the result) and OFS[6:0] are ignored. The same bits are used in 9-bit differential mode because OFS[15] indicates the sign bit, which maps to D[8]. For 16-bit differential mode, OFS[15:0] are directly subtracted from the conversion result data D[15:0]. In 16-bit single-ended mode, there is no field in the OFS corresponding to the least significant result D[0], so odd values, such as -1 or +1, cannot be subtracted from the result.

OFS is automatically set according to calibration requirements once the self-calibration sequence is done, that is, SC3[CAL] is cleared. The user may write to OFS to override the calibration result if desired. If the OFS is written by the user to a value that is different from the calibration value, the ADC error specifications may not be met. Storing the value generated by the calibration function in memory before overwriting with a user-specified value is recommended.

Note

There is an effective limit to the values of offset that can be set by the user. If the magnitude of the offset is too high, the results of the conversions will cap off at the limits.

The offset calibration function may be employed by the user to remove application offsets or DC bias values. OFS may be written with a number in 2's complement format and this offset will be subtracted from the result, or hardware averaged value. To add an offset, store the negative offset in 2's complement format and the effect will be an addition. An offset correction that results in an out-of-range value will be forced to the minimum or maximum value. The minimum value for single-ended conversions is 0x0000; for a differential conversion it is 0x8000.

To preserve accuracy, the calibrated offset value initially stored in OFS must be added to the user-defined offset. For applications that may change the offset repeatedly during operation, store the initial offset calibration value in flash so it can be recovered and added to any user offset adjustment value and the sum stored in OFS.

24.4.8 Temperature sensor

The ADC module includes a temperature sensor whose output is connected to one of the ADC analog channel inputs.

The following equation provides an approximate transfer function of the temperature sensor.

$$\mathrm{Temp} = 25 - \left(\left(V_{\mathrm{TEMP}} - V_{\mathrm{TEMP25}}\right) \div m\right)$$

Figure 24-63. Approximate transfer function of the temperature sensor

where:

- V_{TEMP} is the voltage of the temperature sensor channel at the ambient temperature.
- V_{TEMP25} is the voltage of the temperature sensor channel at 25 °C.
- m is referred as temperature sensor slope in the device data sheet. It is the hot or cold voltage versus temperature slope in V/°C.

For temperature calculations, use the V_{TEMP25} and temperature sensor slope values from the ADC Electricals table.

In application code, the user reads the temperature sensor channel, calculates V_{TEMP}, and compares to V_{TEMP25}. If V_{TEMP} is greater than V_{TEMP25} the cold slope value is applied in the preceding equation. If V_{TEMP} is less than V_{TEMP25}, the hot slope value is applied in the preceding equation. ADC Electricals table may only specify one temperature sensor slope value. In that case, the user could use the same slope for the calculation across the operational temperature range.

For more information on using the temperature sensor, see the application note titled *Temperature Sensor for the HCS08 Microcontroller Family* (document AN3031).

24.4.9 MCU wait mode operation

Wait mode is a lower-power consumption Standby mode from which recovery is fast because the clock sources remain active.

If a conversion is in progress when the MCU enters Wait mode, it continues until completion. Conversions can be initiated while the MCU is in Wait mode by means of the hardware trigger or if continuous conversions are enabled.

The bus clock, bus clock divided by two; and ADACK are available as conversion clock sources while in Wait mode. The use of ALTCLK as the conversion clock source in Wait is dependent on the definition of ALTCLK for this MCU. See the Chip Configuration information on ALTCLK specific to this MCU.

If the compare and hardware averaging functions are disabled, a conversion complete event sets SC1n[COCO] and generates an ADC interrupt to wake the MCU from Wait mode if the respective ADC interrupt is enabled, that is, when SC1n[AIEN]=1. If the hardware averaging function is enabled, SC1n[COCO] will set, and generate an interrupt if enabled, when the selected number of conversions are completed. If the compare function is enabled, SC1n[COCO] will set, and generate an interrupt if enabled, only if the compare conditions are met. If a single conversion is selected and the compare trigger is not met, the ADC will return to its idle state and cannot wake the MCU from Wait mode unless a new conversion is initiated by the hardware trigger.

24.4.10 MCU Normal Stop mode operation

Stop mode is a low-power consumption Standby mode during which most or all clock sources on the MCU are disabled.

24.4.10.1 Normal Stop mode with ADACK disabled

If the asynchronous clock, ADACK, is not selected as the conversion clock, executing a stop instruction aborts the current conversion and places the ADC in its Idle state. The contents of the ADC registers, including Rn, are unaffected by Normal Stop mode. After exiting from Normal Stop mode, a software or hardware trigger is required to resume conversions.

24.4.10.2 Normal Stop mode with ADACK enabled

If ADACK is selected as the conversion clock, the ADC continues operation during Normal Stop mode. See the chip configuration chapter for configuration information for this MCU.

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.

24.4.11 MCU Low-Power Stop mode operation

The ADC module is automatically disabled when the MCU enters Low-Power Stop mode.

All module registers contain their reset values following exit from Low-Power Stop mode. Therefore, the module must be re-enabled and re-configured following exit from Low-Power Stop mode.

NOTE

For the chip specific modes of operation, see the power management information for the device.

24.5 Initialization information

This section gives an example that provides some basic direction on how to initialize and configure the ADC module.

The user can configure the module for 16-bit, 12-bit, 10-bit, or 8-bit single-ended resolution or 16-bit, 13-bit, 11-bit, or 9-bit differential resolution, single or continuous conversion, and a polled or interrupt approach, among many other options. For information used in this example, refer to Table 24-73, Table 24-74, and Table 24-75.

Note

Hexadecimal values are designated by a preceding 0x, binary values designated by a preceding %, and decimal values have no preceding character.

24.5.1 ADC module initialization example

24.5.1.1 Initialization sequence

Before the ADC module can be used to complete conversions, an initialization procedure must be performed. A typical sequence is:

1. Calibrate the ADC by following the calibration instructions in Calibration function.
2. Update CFG to select the input clock source and the divide ratio used to generate ADCK. This register is also used for selecting sample time and low-power configuration.
3. Update SC2 to select the conversion trigger, hardware or software, and compare function options, if enabled.
4. Update SC3 to select whether conversions will be continuous or completed only once (ADCO) and whether to perform hardware averaging.
5. Update SC1:SC1n registers to select whether conversions will be single-ended or differential and to enable or disable conversion complete interrupts. Also, select the input channel which can be used to perform conversions.

24.5.1.2 Pseudo-code example

In this example, the ADC module is set up with interrupts enabled to perform a single 10-bit conversion at low-power with a long sample time on input channel 1, where ADCK is derived from the bus clock divided by 1.

CFG1 = 0x98 (%10011000)

```
        Bit 7   ADLPC   1      Configures for low power, lowers maximum clock speed.
        Bit 6:5 ADIV    00     Sets the ADCK to the input clock ÷ 1.
        Bit 4   ADLSMP  1      Configures for long sample time.
        Bit 3:2   MODE   10    Selects the single-ended 10-bit conversion, differential 11-
bit conversion.
        Bit 1:0   ADICLK   00    Selects the bus clock.
```

SC2 = 0x00 (%00000000)

```
        Bit 7   ADACT   0       Flag indicates if a conversion is in progress.
        Bit 6   ADTRG   0       Software trigger selected.
        Bit 5   ACFE    0       Compare function disabled.
        Bit 4   ACFGT   0       Not used in this example.
        Bit 3   ACREN   0      Compare range disabled.
 Bit 2 DMAEN 0 DMA request disabled.
        Bit 1:0 REFSEL  00     Selects default voltage reference pin pair (External pins VREFH
and VREFL).
```

SC1A = 0x41 (%01000001)

```
        Bit 7   COCO   0       Read-only flag which is set when a conversion completes.
        Bit 6   AIEN   1       Conversion complete interrupt enabled.
 Bit 5 DIFF 0 Single-ended conversion selected.
        Bit 4:0 ADCH   00001   Input channel 1 selected as ADC input channel.
```

RA = 0xxx

```
        Holds results of conversion.
```

CV = 0xxx

```
        Holds compare value when compare function enabled.
```

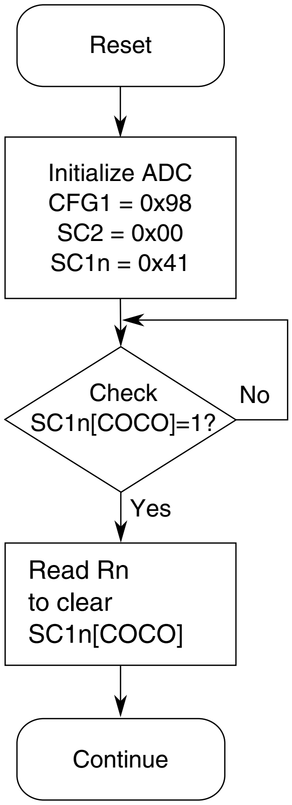


Figure 24-64. Initialization flowchart example

24.6 Application information

The ADC has been designed to be integrated into a microcontroller for use in embedded control applications requiring an ADC.

For guidance on selecting optimum external component values and converter parameters see AN4373: Cookbook for SAR ADC Measurements.

24.6.1 External pins and routing

24.6.1.1 Analog supply pins

Depending on the device, the analog power and ground supplies, V_{DDA} and V_{SSA}, of the ADC module are available as:

- V_{DDA} and V_{SSA} available as separate pins—When available on a separate pin, both V_{DDA} and V_{SSA} must be connected to the same voltage potential as their corresponding MCU digital supply, V_{DD} and V_{SS}, and must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.
- V_{SSA} is shared on the same pin as the MCU digital V_{SS}.
- V_{SSA} and V_{DDA} are shared with the MCU digital supply pins—In these cases, there are separate pads for the analog supplies bonded to the same pin as the corresponding digital supply so that some degree of isolation between the supplies is maintained.

If separate power supplies are used for analog and digital power, the ground connection between these supplies must be at the V_{SSA} pin. This must be the only ground connection between these supplies, if possible. V_{SSA} makes a good single point ground location.

24.6.1.2 Analog voltage reference pins

In addition to the analog supplies, the ADC module has connections for two reference voltage inputs used by the converter:

- V_{REFSH} is the high reference voltage for the converter.
- V_{REFSL} is the low reference voltage for the converter.

The ADC can be configured to accept one of two voltage reference pairs for V_{REFSH} and V_{REFSL}. Each pair contains a positive reference and a ground reference. The two pairs are external, V_{REFH} and V_{REFL} and alternate, V_{ALTH} and V_{ALTL}. These voltage references are selected using SC2[REFSEL]. The alternate voltage reference pair, V_{ALTH} and V_{ALTL}, may select additional external pins or internal sources based on MCU configuration. See the chip configuration information on the voltage references specific to this MCU.

In some packages, the external or alternate pairs are connected in the package to V_{DDA} and V_{SSA}, respectively. One of these positive references may be shared on the same pin as V_{DDA} on some devices. One of these ground references may be shared on the same pin as V_{SSA} on some devices.

If externally available, the positive reference may be connected to the same potential as V_{DDA} or may be driven by an external source to a level between the minimum Ref Voltage High and the V_{DDA} potential. The positive reference must never exceed V_{DDA}. If externally available, the ground reference must be connected to the same voltage potential as V_{SSA}. The voltage reference pairs must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.

AC current in the form of current spikes required to supply charge to the capacitor array at each successive approximation step is drawn through the V_{REFH} and V_{REFL} loop. The best external component to meet this current demand is a 0.1 μF capacitor with good high-frequency characteristics. This capacitor is connected between V_{REFH} and V_{REFL} and must be placed as near as possible to the package pins. Resistance in the path is not recommended because the current causes a voltage drop that could result in conversion errors. Inductance in this path must be minimum, that is, parasitic only.

24.6.1.3 Analog input pins

The external analog inputs are typically shared with digital I/O pins on MCU devices.

Empirical data shows that capacitors on the analog inputs improve performance in the presence of noise or when the source impedance is high. Use of 0.01 μF capacitors with good high-frequency characteristics is sufficient. These capacitors are not necessary in all cases, but when used, they must be placed as near as possible to the package pins and be referenced to V_{SSA}.

For proper conversion, the input voltage must fall between V_{REFH} and V_{REFL}. If the input is equal to or exceeds V_{REFH}, the converter circuit converts the signal to 0xFFF, which is full scale 12-bit representation, 0x3FF, which is full scale 10-bit representation, or 0xFF, which is full scale 8-bit representation. If the input is equal to or less than V_{REFL}, the converter circuit converts it to 0x000. Input voltages between V_{REFH} and V_{REFL} are straight-line linear conversions. There is a brief current associated with V_{REFL} when the sampling capacitor is charging.

For minimal loss of accuracy due to current injection, pins adjacent to the analog input pins must not be transitioning during conversions.

24.6.2 Sources of error

24.6.2.1 Sampling error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy.

$$RAS + RADIN = SC / (FMAX * NUMTAU * CADIN)$$

Figure 24-65. Sampling equation

Where:

RAS = External analog source resistance

SC = Number of ADCK cycles used during sample window

CADIN = Internal ADC input capacitance

NUMTAU = $-\ln(LSBERR / 2^N)$

LSBERR = value of acceptable sampling error in LSBs

N = 8 in 8-bit mode, 10 in 10-bit mode, 12 in 12-bit mode or 16 in 16-bit mode

Higher source resistances or higher-accuracy sampling is possible by setting CFG1[ADLSMP] and changing CFG2[ADLSTS] to increase the sample window, or decreasing ADCK frequency to increase sample time.

24.6.2.2 Pin leakage error

Leakage on the I/O pins can cause conversion error if the external analog source resistance, R_{AS}, is high. If this error cannot be tolerated by the application, keep R_{AS} lower than $V_{REFH} / (4 \times I_{LEAK} \times 2^N)$ for less than 1/4 LSB leakage error, where N = 8 in 8-bit mode, 10 in 10-bit mode, 12 in 12-bit mode, or 16 in 16-bit mode.

24.6.2.3 Noise-induced errors

System noise that occurs during the sample or conversion process can affect the accuracy of the conversion. The ADC accuracy numbers are guaranteed as specified only if the following conditions are met:

- There is a 0.1 μF low-ESR capacitor from V_{REFH} to V_{REFL}.
- There is a 0.1 μF low-ESR capacitor from V_{DDA} to V_{SSA}.
- If inductive isolation is used from the primary supply, an additional 1 μF capacitor is placed from V_{DDA} to V_{SSA}.
- V_{SSA}, and V_{REFL}, if connected, is connected to V_{SS} at a quiet point in the ground plane.
- Operate the MCU in Wait or Normal Stop mode before initiating (hardware-triggered conversions) or immediately after initiating (hardware- or software-triggered conversions) the ADC conversion.

- For software triggered conversions, immediately follow the write to SC1 with a Wait instruction or Stop instruction.
- For Normal Stop mode operation, select ADACK as the clock source. Operation in Normal Stop reduces V_{DD} noise but increases effective conversion time due to stop recovery.

- There is no I/O switching, input or output, on the MCU during the conversion.

There are some situations where external system activity causes radiated or conducted noise emissions or excessive V_{DD} noise is coupled into the ADC. In these situations, or when the MCU cannot be placed in Wait or Normal Stop mode, or I/O activity cannot be halted, the following actions may reduce the effect of noise on the accuracy:

- Place a 0.01 μF capacitor (C_{AS}) on the selected input channel to V_{REFL} or V_{SSA}. This improves noise issues, but affects the sample rate based on the external analog source resistance.
- Average the result by converting the analog input many times in succession and dividing the sum of the results. Four samples are required to eliminate the effect of a 1 LSB, one-time error.
- Reduce the effect of synchronous noise by operating off the asynchronous clock, that is, ADACK, and averaging. Noise that is synchronous to ADCK cannot be averaged out.

24.6.2.4 Code width and quantization error

The ADC quantizes the ideal straight-line transfer function into 65536 steps in the 16-bit mode). Each step ideally has the same height, that is, 1 code, and width. The width is defined as the delta between the transition points to one code and the next. The ideal code width for an N-bit converter, where N can be 16, 12, 10, or 8, defined as 1 LSB, is:

$$1\text{LSB} = \left(V_{REFH} \right) / 2^N$$

Figure 24-66. Ideal code width for an N-bit converter

There is an inherent quantization error due to the digitization of the result. For 8-bit, 10-bit, or 12-bit conversions, the code transitions when the voltage is at the midpoint between the points where the straight line transfer function is exactly represented by the actual transfer function. Therefore, the quantization error will be ± 1/2 LSB in 8-bit, 10-bit, or 12-bit modes. As a consequence, however, the code width of the first (0x000) conversion is only 1/2 LSB and the code width of the last (0xFF or 0x3FF) is 1.5 LSB.

For 16-bit conversions, the code transitions only after the full code width is present, so the quantization error is -1 LSB to 0 LSB and the code width of each step is 1 LSB.

24.6.2.5 Linearity errors

The ADC may also exhibit non-linearity of several forms. Every effort has been made to reduce these errors, but the system designers must be aware of these errors because they affect overall accuracy:

- Zero-scale error (E_{ZS}), sometimes called offset: This error is defined as the difference between the actual code width of the first conversion and the ideal code width. This is 1/2 LSB in 8-bit, 10-bit, or 12-bit modes and 1 LSB in 16-bit mode. If the first conversion is 0x001, the difference between the actual 0x001 code width and its ideal (1 LSB) is used.
- Full-scale error (E_{FS}): This error is defined as the difference between the actual code width of the last conversion and the ideal code width. This is 1.5 LSB in 8-bit, 10-bit, or 12-bit modes and 1 LSB in 16-bit mode. If the last conversion is 0x3FE, the difference between the actual 0x3FE code width and its ideal (1 LSB) is used.
- Differential non-linearity (DNL): This error is defined as the worst-case difference between the actual code width and the ideal code width for all conversions.
- Integral non-linearity (INL): This error is defined as the highest-value or absolute value that the running sum of DNL achieves. More simply, this is the worst-case difference of the actual transition voltage to a given code and its corresponding ideal transition voltage, for all codes.
- Total unadjusted error (TUE): This error is defined as the difference between the actual transfer function and the ideal straight-line transfer function and includes all forms of error.

24.6.2.6 Code jitter, non-monotonicity, and missing codes

Analog-to-digital converters are susceptible to three special forms of error:

- Code jitter: Code jitter is when, at certain points, a given input voltage converts to one of the two values when sampled repeatedly. Ideally, when the input voltage is infinitesimally smaller than the transition voltage, the converter yields the lower code, and vice-versa. However, even small amounts of system noise can cause the converter to be indeterminate, between two codes, for a range of input voltages around the transition voltage.

This error may be reduced by repeatedly sampling the input and averaging the result. Additionally, the techniques discussed in Noise-induced errors reduces this error.

- Non-monotonicity: Non-monotonicity is defined as when, except for code jitter, the converter converts to a lower code for a higher input voltage.
- Missing codes: Missing codes are those values never converted for any input value.

In 8-bit or 10-bit mode, the ADC is guaranteed to be monotonic and have no missing codes.

Chapter 25
Comparator (CMP)

25.1 Introduction

NOTE

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

The comparator (CMP) module provides a circuit for comparing two analog input voltages. The comparator circuit is designed to operate across the full range of the supply voltage, known as rail-to-rail operation.

The Analog MUX (ANMUX) provides a circuit for selecting an analog input signal from eight channels. One signal is provided by the 6-bit digital-to-analog converter (DAC). The mux circuit is designed to operate across the full range of the supply voltage.

The 6-bit DAC is 64-tap resistor ladder network which provides a selectable voltage reference for applications where voltage reference is needed. The 64-tap resistor ladder network divides the supply reference V_{in} into 64 voltage levels. A 6-bit digital signal input selects the output voltage level, which varies from V_{in} to $V_{in}/64$. V_{in} can be selected from two voltage sources, V_{in1} and V_{in2}. The 6-bit DAC from a comparator is available as an on-chip internal signal only and is not available externally to a pin.

25.1.1 CMP features

The CMP has the following features:

- Operational over the entire supply range
- Inputs may range from rail to rail
- Programmable hysteresis control

- Selectable interrupt on rising-edge, falling-edge, or both rising or falling edges of the comparator output
- Selectable inversion on comparator output
- Capability to produce a wide range of outputs such as:
  - Sampled
  - Digitally filtered:
    - Filter can be bypassed
    - Can be clocked via scaled bus clock
- External hysteresis can be used at the same time that the output filter is used for internal functions
- Two software selectable performance levels:
  - Shorter propagation delay at the expense of higher power
  - Low power, with longer propagation delay
- DMA transfer support
  - A comparison event can be selected to trigger a DMA transfer
- Functional in all modes of operation
- The filter functions are not available in the following modes:
  - Stop
  - VLPS
  - LLS

25.1.2 6-bit DAC key features

- 6-bit resolution
- Selectable supply reference source
- Power Down mode to conserve power when not in use
- Option to route the output to internal comparator input

25.1.3 ANMUX key features

- Two 8-to-1 channel mux
- Operational over the entire supply range

25.1.4 CMP, DAC and ANMUX diagram

The following figure shows the block diagram for the High-Speed Comparator, DAC, and ANMUX modules.

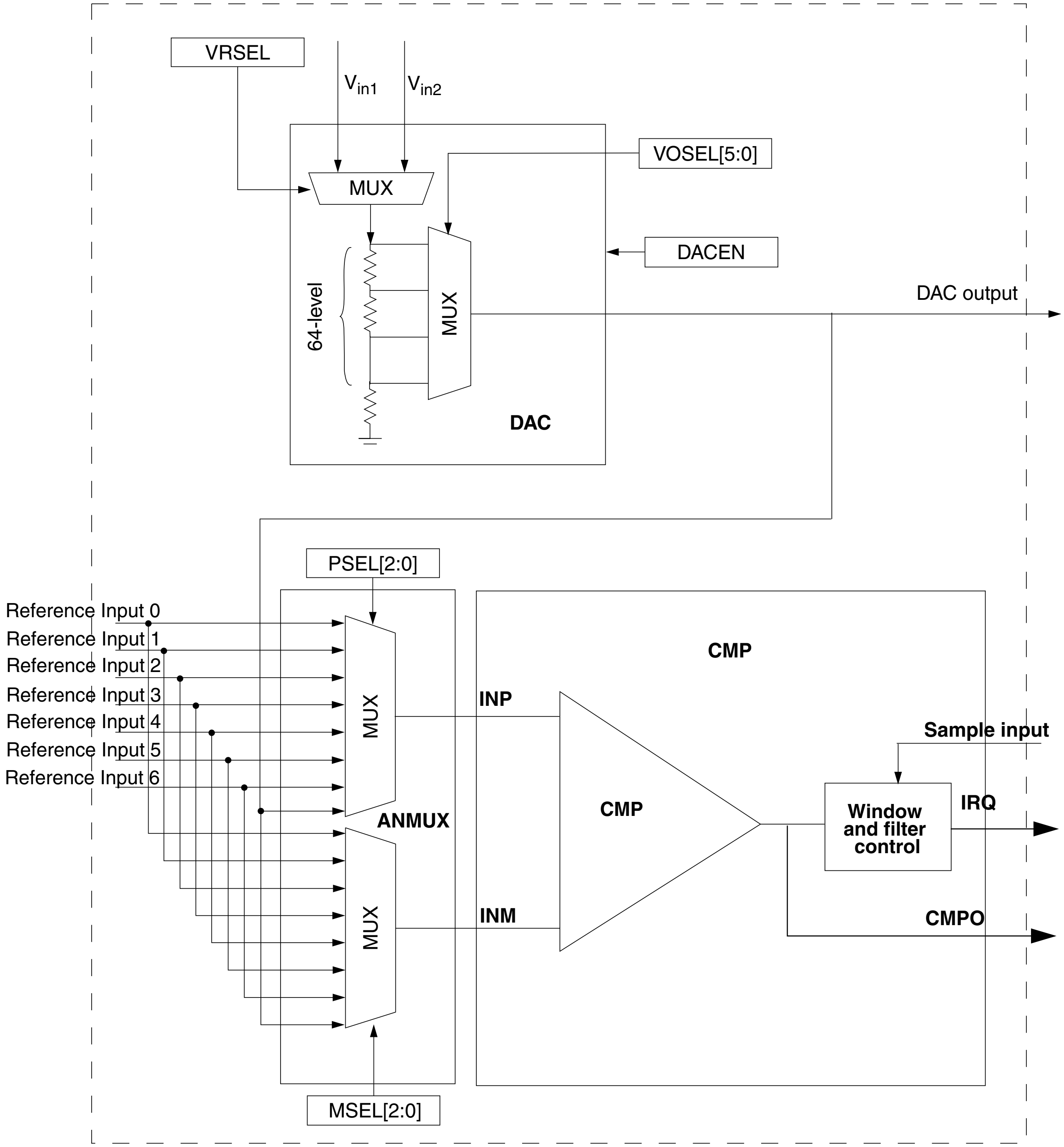


Figure 25-1. CMP, DAC and ANMUX block diagram

25.1.5 CMP block diagram

The following figure shows the block diagram for the CMP module.

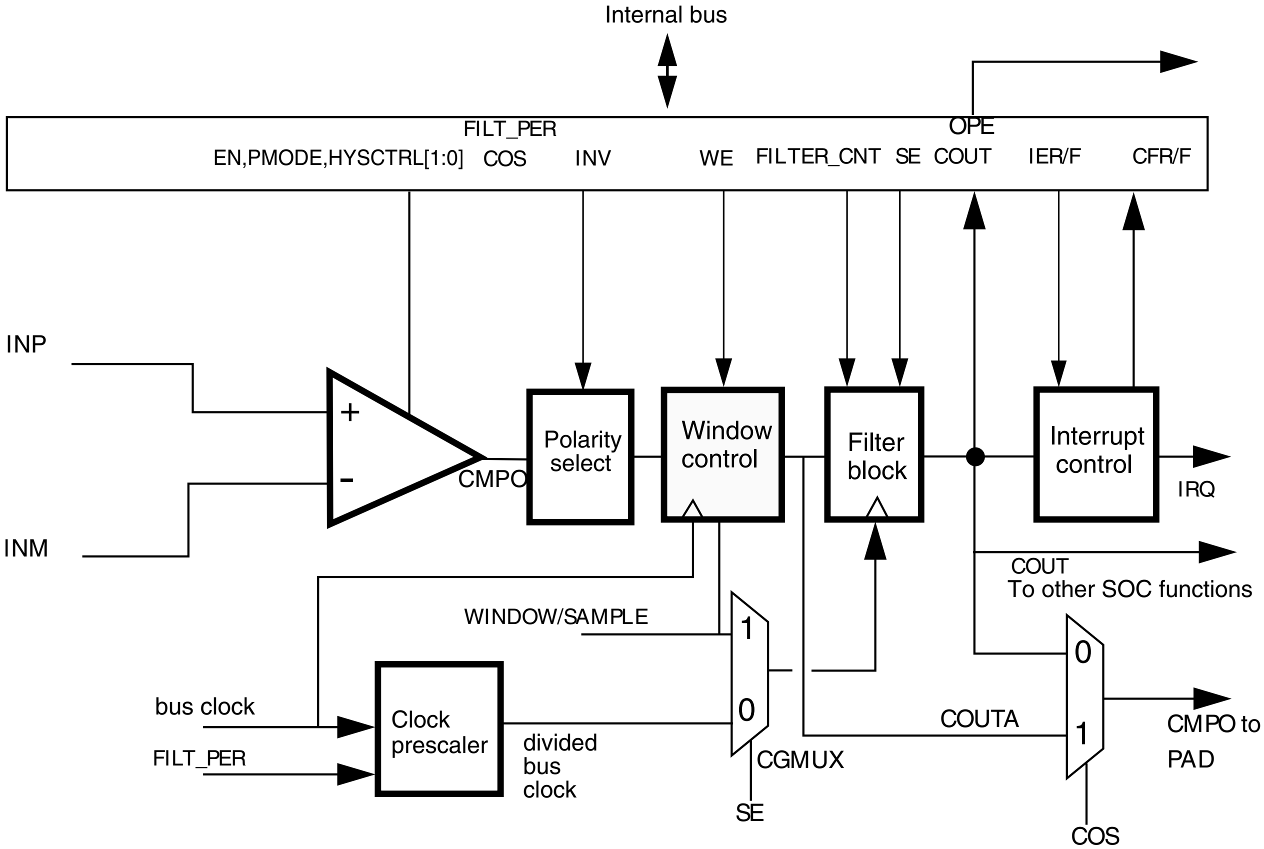


Figure 25-2. Comparator module block diagram

In the CMP block diagram:

- The Window Control block is bypassed when CR1[WE] = 0
- The Filter block is bypassed when not in use.
- The Filter block acts as a simple sampler if the filter is bypassed and CR0[FILTER_CNT] is set to 0x01.
- The Filter block filters based on multiple samples when the filter is bypassed and CR0[FILTER_CNT] is set greater than 0x01.
    - CR1[SE] = 0, the divided bus clock is used as sampling clock
- If enabled, the Filter block will incur up to one bus clock additional latency penalty on COUT due to the fact that COUT, which is crossing clock domain boundaries, must be resynchronized to the bus clock.

25.2 Memory map/register definitions

CMP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_3000	CMP Control Register 0 (CMP0_CR0)	8	R/W	00h	25.2.1/475
4007_3001	CMP Control Register 1 (CMP0_CR1)	8	R/W	00h	25.2.2/476
4007_3002	CMP Filter Period Register (CMP0_FPR)	8	R/W	00h	25.2.3/477
4007_3003	CMP Status and Control Register (CMP0_SCR)	8	R/W	00h	25.2.4/478
4007_3004	DAC Control Register (CMP0_DACCR)	8	R/W	00h	25.2.5/479
4007_3005	MUX Control Register (CMP0_MUXCR)	8	R/W	00h	25.2.6/479

25.2.1 CMP Control Register 0 (CMPx_CR0)

Address: 4007_3000h base + 0h offset = 4007_3000h

Bit	7	6	5	4	3	2	1	0
Read	0	FILTER_CNT			0	0	HYSTCTR	
Write								
Reset	0	0	0	0	0	0	0	0

CMPx_CR0 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 FILTER_CNT	Filter Sample Count Represents the number of consecutive samples that must agree prior to the comparator ouput filter accepting a new output state. For information regarding filter programming and latency, see the Functional description. 000 Filter is disabled. SE = 0, COUT = COUTA. 001 One sample must agree. The comparator output is simply sampled. 010 2 consecutive samples must agree. 011 3 consecutive samples must agree. 100 4 consecutive samples must agree. 101 5 consecutive samples must agree. 110 6 consecutive samples must agree. 111 7 consecutive samples must agree.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

CMPx_CR0 field descriptions (continued)

Field	Description
1–0 HYSTCTR	Comparator hard block hysteresis control Defines the programmable hysteresis level. The hysteresis values associated with each level are device-specific. See the Data Sheet of the device for the exact values. 00 Level 0 01 Level 1 10 Level 2 11 Level 3

25.2.2 CMP Control Register 1 (CMPx_CR1)

Address: 4007_3000h base + 1h offset = 4007_3001h

Bit	7	6	5	4	3	2	1	0
Read Write	SE	WE	TRIGM	PMODE	INV	COS	OPE	EN
Reset	0	0	0	0	0	0	0	0

CMPx_CR1 field descriptions

Field	Description
7 SE	Sample Enable SE must be clear to 0 and usage of sample operation is limited to a divided version of the bus clock. 0 Sampling mode is not selected. 1 Sampling mode is selected.
6 WE	Windowing Enable The CMP does not support window compare function and a 0 must always be written to WE. 0 Windowing mode is not selected. 1 Windowing mode is selected.
5 TRIGM	Trigger Mode Enable CMP and DAC are configured to CMP Trigger mode when CMP_CR1[TRIGM] is set to 1. In addition, the CMP should be enabled. If the DAC is to be used as a reference to the CMP, it should also be enabled. CMP Trigger mode depends on an external timer resource to periodically enable the CMP and 6-bit DAC in order to generate a triggered compare. Upon setting TRIGM, the CMP and DAC are placed in a standby state until an external timer resource trigger is received. See the chip configuration for details about the external timer resource. 0 Trigger mode is disabled. 1 Trigger mode is enabled.
4 PMODE	Power Mode Select

Table continues on the next page...

CMPx_CR1 field descriptions (continued)

Field	Description
	See the electrical specifications table in the device Data Sheet for details. 0 Low-Speed (LS) Comparison mode selected. In this mode, CMP has slower output propagation delay and lower current consumption. 1 High-Speed (HS) Comparison mode selected. In this mode, CMP has faster output propagation delay and higher current consumption.
3 INV	Comparator INVERT Allows selection of the polarity of the analog comparator function. It is also driven to the COUT output, on both the device pin and as SCR[COUT], when OPE=0. 0 Does not invert the comparator output. 1 Inverts the comparator output.
2 COS	Comparator Output Select 0 Set the filtered comparator output (CMPO) to equal COUT. 1 Set the unfiltered comparator output (CMPO) to equal COUTA.
1 OPE	Comparator Output Pin Enable 0 CMPO is not available on the associated CMPO output pin. If the comparator does not own the pin, this field has no effect. 1 CMPO is available on the associated CMPO output pin. The comparator output (CMPO) is driven out on the associated CMPO output pin if the comparator owns the pin. If the comparator does not own the field, this bit has no effect.
0 EN	Comparator Module Enable Enables the Analog Comparator module. When the module is not enabled, it remains in the off state, and consumes no power. When the user selects the same input from analog mux to the positive and negative port, the comparator is disabled automatically. 0 Analog Comparator is disabled. 1 Analog Comparator is enabled.

25.2.3 CMP Filter Period Register (CMPx_FPR)

Address: 4007_3000h base + 2h offset = 4007_3002h

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

CMPx_FPR field descriptions

Field	Description
7–0 FILT_PER	Filter Sample Period Specifies the sampling period, in bus clock cycles, of the comparator output filter, when CR1[SE]=0. Setting FILT_PER to 0x0 disables the filter. Filter programming and latency details appear in the Functional description.

25.2.4 CMP Status and Control Register (CMPx_SCR)

Address: 4007_3000h base + 3h offset = 4007_3003h

Bit	7	6	5	4	3	2	1	0
Read	0	DMAEN	0	IER	IEF	CFR	CFF	COUT
Write						w1c	w1c	
Reset	0	0	0	0	0	0	0	0

CMPx_SCR field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 DMAEN	DMA Enable Control Enables the DMA transfer triggered from the CMP module. When this field is set, a DMA request is asserted when CFR or CFF is set. 0 DMA is disabled. 1 DMA is enabled.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 IER	Comparator Interrupt Enable Rising Enables the CFR interrupt from the CMP. When this field is set, an interrupt will be asserted when CFR is set. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IEF	Comparator Interrupt Enable Falling Enables the CFF interrupt from the CMP. When this field is set, an interrupt will be asserted when CFF is set. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 CFR	Analog Comparator Flag Rising Detects a rising-edge on COUT, when set, during normal operation. CFR is cleared by writing 1 to it. During Stop modes, CFR is level sensitive . 0 Rising-edge on COUT has not been detected. 1 Rising-edge on COUT has occurred.
1 CFF	Analog Comparator Flag Falling Detects a falling-edge on COUT, when set, during normal operation. CFF is cleared by writing 1 to it. During Stop modes, CFF is level sensitive . 0 Falling-edge on COUT has not been detected. 1 Falling-edge on COUT has occurred.

Table continues on the next page...

CMPx_SCR field descriptions (continued)

Field	Description
0 COUT	Analog Comparator Output Returns the current value of the Analog Comparator output, when read. The field is reset to 0 and will read as CR1[INV] when the Analog Comparator module is disabled, that is, when CR1[EN] = 0. Writes to this field are ignored.

25.2.5 DAC Control Register (CMPx_DACCR)

Address: 4007_3000h base + 4h offset = 4007_3004h

Bit	7	6	5	4	3	2	1	0
Read Write	DACEN	VRSEL	VOSEL					
Reset	0	0	0	0	0	0	0	0

CMPx_DACCR field descriptions

Field	Description
7 DACEN	DAC Enable Enables the DAC. When the DAC is disabled, it is powered down to conserve power. 0 DAC is disabled. 1 DAC is enabled.
6 VRSEL	Supply Voltage Reference Source Select 0 V_{in1} is selected as resistor ladder network supply reference. 1 V_{in2} is selected as resistor ladder network supply reference.
5–0 VOSEL	DAC Output Voltage Select Selects an output voltage from one of 64 distinct levels. `DACO = (V in /64) * (VOSEL[5:0] + 1)` , so the DACO range is from V_{in} /64 to V_{in} .

25.2.6 MUX Control Register (CMPx_MUXCR)

Address: 4007_3000h base + 5h offset = 4007_3005h

Bit	7	6	5	4	3	2	1	0
Read	PSTM	0	PSEL			MSEL		
Write								
Reset	0	0	0	0	0	0	0	0

CMPx_MUXCR field descriptions

Field	Description
7 PSTM	Pass Through Mode Enable

Table continues on the next page...

CMPx_MUXCR field descriptions (continued)

Field	Description
	This bit is used to enable to MUX pass through mode. Pass through mode is always available but for some devices this feature must be always disabled due to the lack of package pins. 0 Pass Through Mode is disabled. 1 Pass Through Mode is enabled.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–3 PSEL	Plus Input Mux Control Determines which input is selected for the plus input of the comparator. For INx inputs, see CMP, DAC, and ANMUX block diagrams. **NOTE:** When an inappropriate operation selects the same input for both muxes, the comparator automatically shuts down to prevent itself from becoming a noise generator. 000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7
2–0 MSEL	Minus Input Mux Control Determines which input is selected for the minus input of the comparator. For INx inputs, see CMP, DAC, and ANMUX block diagrams. **NOTE:** When an inappropriate operation selects the same input for both muxes, the comparator automatically shuts down to prevent itself from becoming a noise generator. 000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7

25.3 Functional description

The CMP module can be used to compare two analog input voltages applied to INP and INM.

CMPO is high when the non-inverting input is greater than the inverting input, and is low when the non-inverting input is less than the inverting input. This signal can be selectively inverted by setting CR1[INV] = 1.

SCR[IER] and SCR[IEF] are used to select the condition which will cause the CMP module to assert an interrupt to the processor. SCR[CFF] is set on a falling-edge and SCR[CFR] is set on rising-edge of the comparator output. The optionally filtered CMPO can be read directly through SCR[COUT].

25.3.1 CMP functional modes

There are the following main sub-blocks to the CMP module:

- The comparator itself
- The filter function

The filter, CR0[FILTER_CNT], can be clocked from an internal clock source only. The filter is programmable with respect to the number of samples that must agree before a change in the output is registered. In the simplest case, only one sample must agree. In this case, the filter acts as a simple sampler.

The comparator filter and sampling features can be combined as shown in the following table. Individual modes are discussed below.

Table 25-15. Comparator sample/filter controls

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation
1	0	X	X	X	X	**Disabled** See the Disabled mode (# 1).
2A	1	0	0	0x00	X	**Continuous Mode** See the Continuous mode (#s 2A & 2B).
2B	1	0	0	X	0x00	
3B	1	0	0	0x01	> 0x00	**Sampled, Non-Filtered mode** See the Sampled, Non-Filtered mode (#s 3B).
4B	1	0	0	> 0x01	> 0x00	**Sampled, Filtered mode** See the Sampled, Filtered mode (#s 4B).
All other combinations of CR1[EN], CR1[WE], CR1[SE], CR0[FILTER_CNT], and FPR[FILT_PER] are illegal.						

For cases where a comparator is used to drive a fault input, for example, for a , it must be configured to operate in Continuous mode so that an external fault can immediately pass through the comparator to the target fault circuitry.

Note

Filtering and sampling settings must be changed only after setting CR1[SE]=0 and CR0[FILTER_CNT]=0x00. This resets the filter to a known state.

25.3.1.1 Disabled mode (# 1)

In Disabled mode, the analog comparator is non-functional and consumes no power. CMPO is 0 in this mode.

25.3.1.2 Continuous mode (#s 2A & 2B)

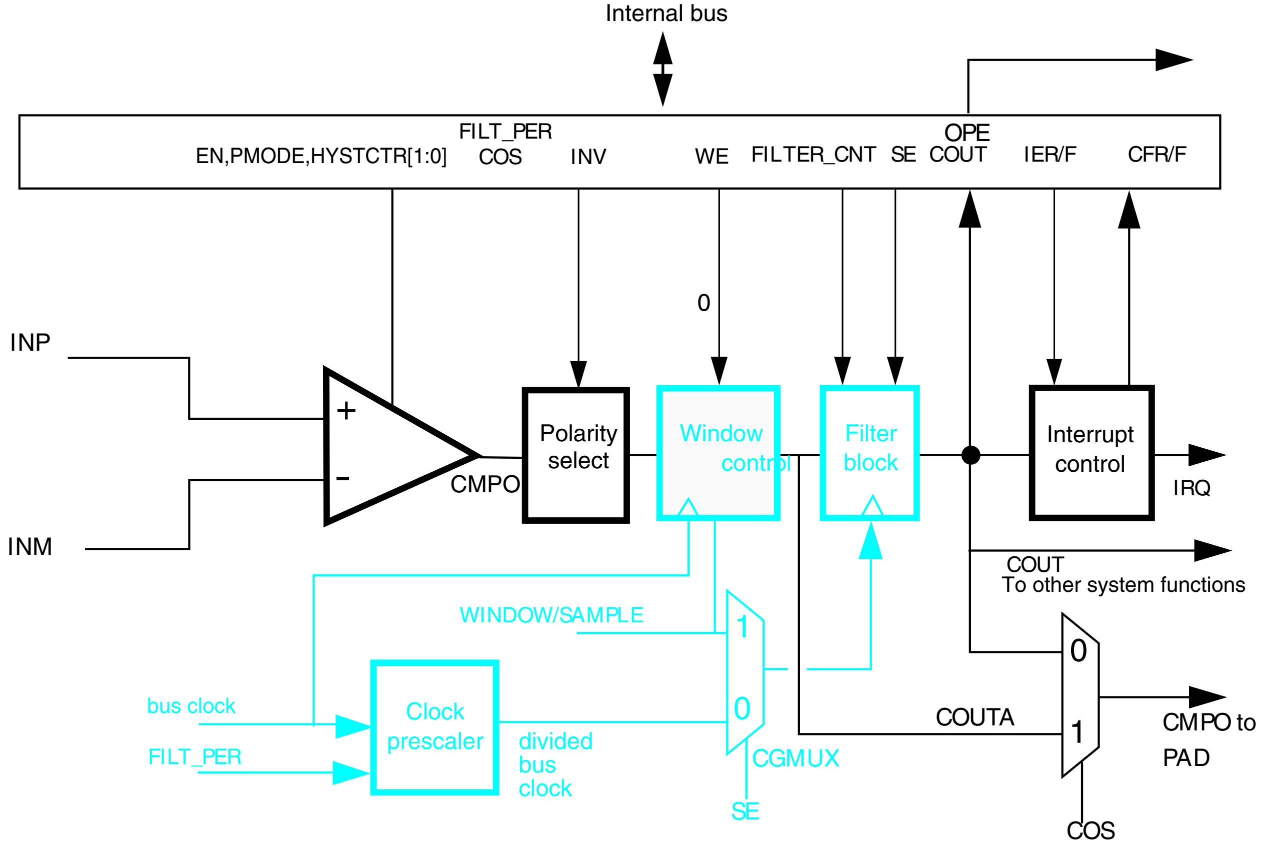


Figure 25-15. Comparator operation in Continuous mode

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. SCR[COUT] is updated continuously. The path from comparator input pins to output pin is operating in combinational unclocked mode. COUT and COUTA are identical.

For control configurations which result in disabling the filter block, see the Filter Block Bypass Logic diagram.

25.3.1.3 Sampled, Non-Filtered mode (#s 3B)

In Sampled, Non-Filtered mode, the analog comparator block is powered and active. The path from analog inputs to COUTA is combinational unclocked. Windowing control is completely bypassed. COUTA is sampled whenever a rising-edge is detected on the filter block clock input.

The comparator filter has no other function than sample/hold of the comparator output in this mode (# 3B).

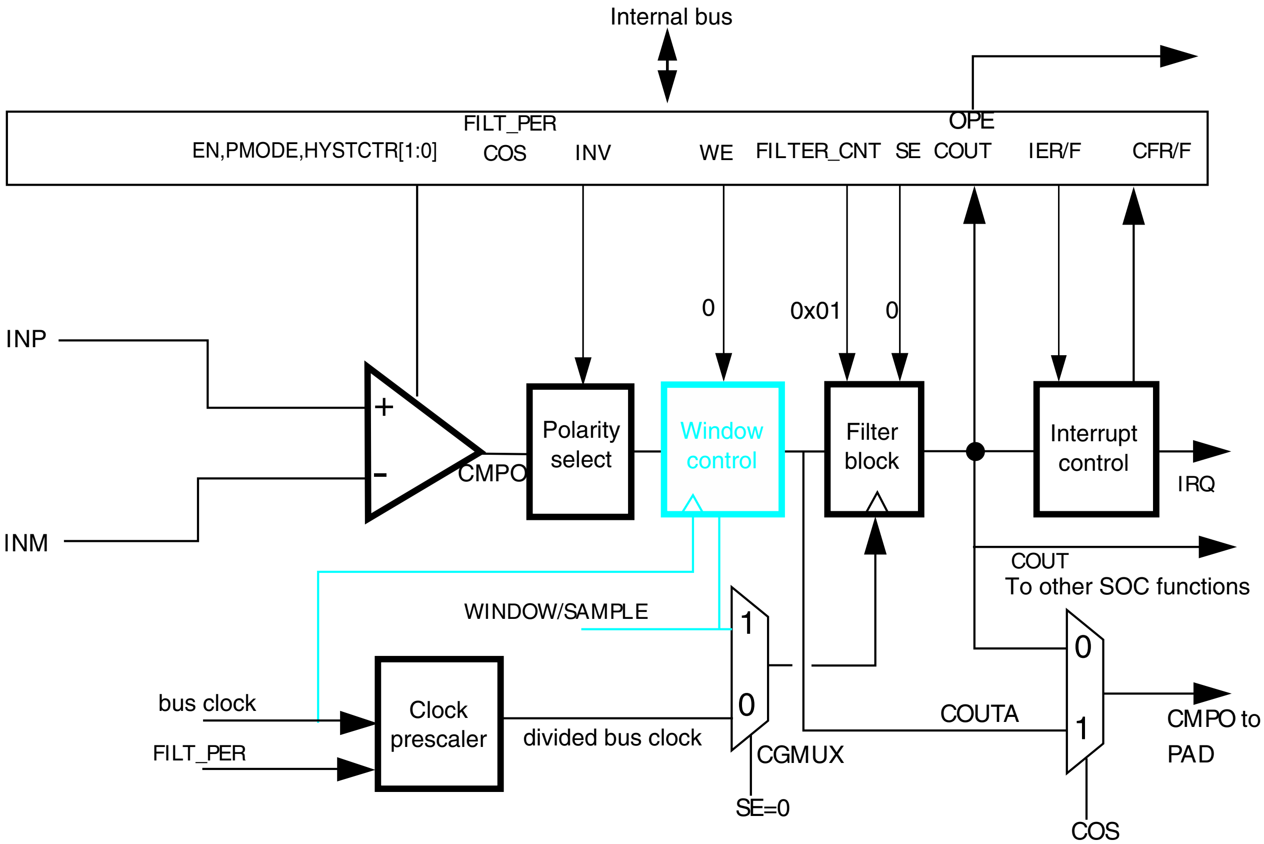


Figure 25-16. Sampled, Non-Filtered (# 3B): sampling interval internally derived

25.3.1.4 Sampled, Filtered mode (#s 4B)

In Sampled, Filtered mode, the analog comparator block is powered and active. The path from analog inputs to COUTA is combinational unclocked. Windowing control is completely bypassed. COUTA is sampled whenever a rising edge is detected on the filter block clock input.

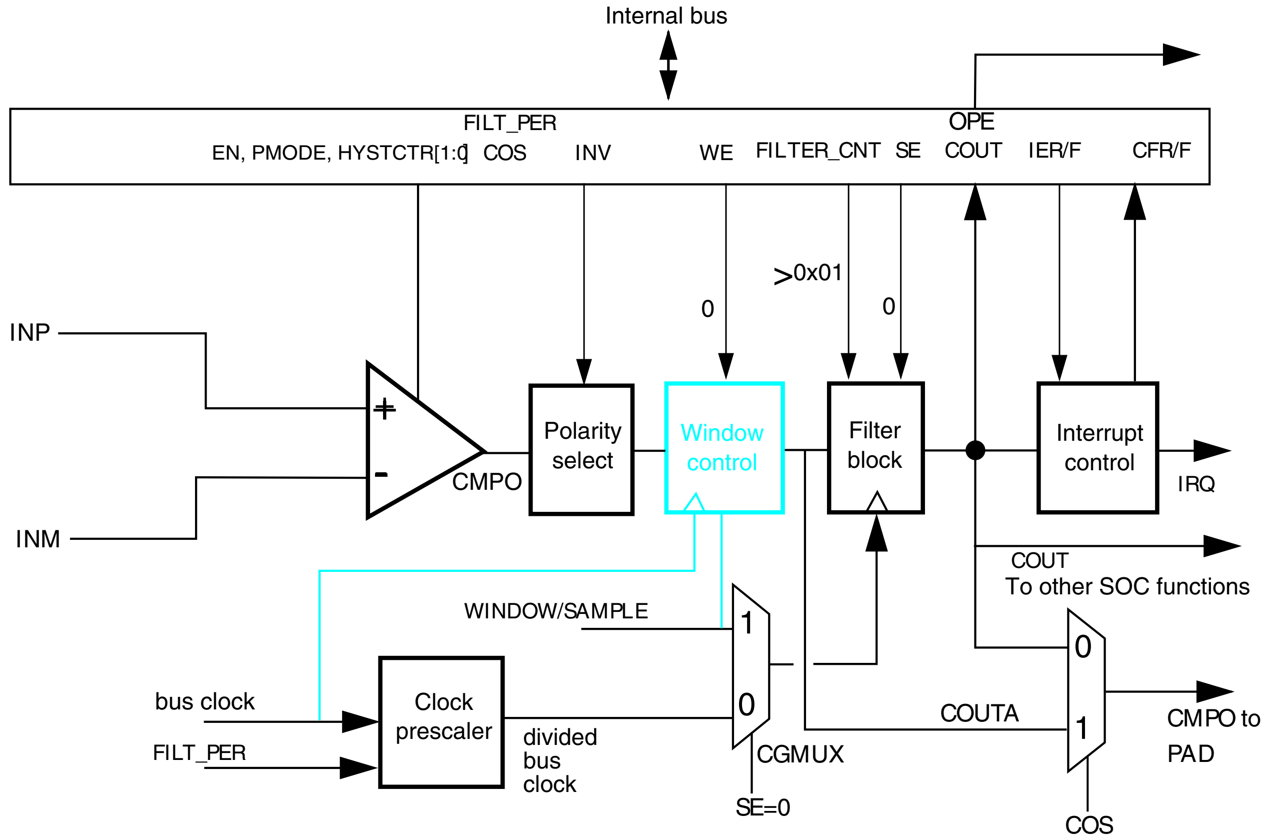


Figure 25-17. Sampled, Filtered (# 4B): sampling point internally derived

The only difference in operation between Sampled, Non-Filtered (# 3B) and Sampled, Filtered (# 4B) is that now, CR0[FILTER_CNT]>1, which activates filter operation.

25.3.2 Power modes

25.3.2.1 Wait mode operation

During Wait and VLPW modes, the CMP, if enabled, continues to operate normally and a CMP interrupt can wake the MCU.

25.3.2.2 Stop mode operation

Depending on clock restrictions related to the MCU core or core peripherals, the MCU is brought out of stop when a compare event occurs and the corresponding interrupt is enabled. Similarly, if CR1[OPE] is enabled, the comparator output operates as in the normal operating mode and comparator output is placed onto the external pin. In Stop modes, the comparator can be operational in both:

- High-Speed (HS) Comparison mode when CR1[PMODE] = 1
- Low-Speed (LS) Comparison mode when CR1[PMODE] = 0

It is recommended to use the LS mode to minimize power consumption.

If stop is exited with a reset, all comparator registers are put into their reset state.

25.3.2.3 Low-Leakage mode operation

When the chip is in Low-Leakage modes:

- The CMP module is partially functional and is limited to Low-Speed mode, regardless of CR1[PMODE] setting
- Windowed, Sampled, and Filtered modes are not supported
- The CMP output pin is latched and does not reflect the compare output state.

The positive- and negative-input voltage can be supplied from external pins or the DAC output. The MCU can be brought out of the Low-Leakage mode if a compare event occurs and the CMP interrupt is enabled. After wakeup from low-leakage modes, the CMP module is in the reset state except for SCR[CFF] and SCR[CFR].

25.3.2.4 Background Debug Mode Operation

When the microcontroller is in active background debug mode, the CMP continues to operate normally.

25.3.3 Startup and operation

A typical startup sequence is listed here.

- The time required to stabilize COUT will be the power-on delay of the comparators plus the largest propagation delay from a selected analog source through the analog

comparator and filter. See the Data Sheets for power-on delays of the comparators. The filter delay is specified in the Low-pass filter.
- During operation, the propagation delay of the selected data paths must always be considered. It may take many bus clock cycles for COUT and SCR[CFR]/SCR[CFF] to reflect an input change or a configuration change to one of the components involved in the data path.
- When programmed for filtering modes, COUT will initially be equal to 0, until sufficient clock cycles have elapsed to fill all stages of the filter. This occurs even if COUTA is at a logic 1.

25.3.4 Low-pass filter

The low-pass filter operates on the unfiltered and unsynchronized and optionally inverted comparator output COUTA and generates the filtered and synchronized output COUT.

Both COUTA and COUT can be configured as module outputs and are used for different purposes within the system.

Synchronization and edge detection are always used to determine status register bit values. They also apply to COUT for all sampling modes. Filtering can be performed using an internal timebase defined by FPR[FILT_PER] to determine sample time.

The need for digital filtering and the amount of filtering is dependent on user requirements. Filtering can become more useful in the absence of an external hysteresis circuit. Without external hysteresis, high-frequency oscillations can be generated at COUTA when the selected INM and INP input voltages differ by less than the offset voltage of the differential comparator.

25.3.4.1 Enabling filter modes

Filter modes can be enabled by:

- Setting CR0[FILTER_CNT] > 0x01 and
- Setting FPR[FILT_PER] to a nonzero value

Using the divided bus clock to drive the filter, it will take samples of COUTA every FPR[FILT_PER] bus clock cycles.

The filter output will be at logic 0 when first initalized, and will subsequently change when all the consecutive CR0[FILTER_CNT] samples agree that the output value has changed. In other words, SCR[COUT] will be 0 for some initial period, even when COUTA is at logic 1.

Setting FPR[FILT_PER] to 0 disables the filter and eliminates switching current associated with the filtering process.

Note

Always switch to this setting prior to making any changes in filter parameters. This resets the filter to a known state. Switching CR0[FILTER_CNT] on the fly without this intermediate step can result in unexpected behavior.

25.3.4.2 Latency issues

The value of FPR[FILT_PER] or SAMPLE period must be set such that the sampling period is just longer than the period of the expected noise. This way a noise spike will corrupt only one sample. The value of CR0[FILTER_CNT] must be chosen to reduce the probability of noisy samples causing an incorrect transition to be recognized. The probability of an incorrect transition is defined as the probability of an incorrect sample raised to the power of CR0[FILTER_CNT].

The values of FPR[FILT_PER] or SAMPLE period and CR0[FILTER_CNT] must also be traded off against the desire for minimal latency in recognizing actual comparator output transitions. The probability of detecting an actual output change within the nominal latency is the probability of a correct sample raised to the power of CR0[FILTER_CNT].

The following table summarizes maximum latency values for the various modes of operation *in the absence of noise*. Filtering latency is restarted each time an actual output transition is masked by noise.

Table 25-16. Comparator sample/filter maximum latencies

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation	Maximum latency[1]
1	0	X	X	X	X	Disabled	N/A
2A	1	0	0	0x00	X	Continuous Mode	T_{PD}
2B	1	0	0	X	0x00		
3B	1	0	0	0x01	> 0x00	Sampled, Non-Filtered mode	T_{PD} + (FPR[FILT_PER] * T_{per}) + T_{per}
4B	1	0	0	> 0x01	> 0x00	Sampled, Filtered mode	T_{PD} + (CR0[FILTER_CNT] * FPR[FILT_PER] x T_{per}) + T_{per}

1. T_{PD} represents the intrinsic delay of the analog component plus the polarity select logic. T_{per} is the period of the bus clock.

25.4 CMP interrupts

The CMP module is capable of generating an interrupt on either the rising- or falling-edge of the comparator output, or both.

The following table gives the conditions in which the interrupt request is asserted and deasserted.

When	Then
SCR[IER] and SCR[CFR] are set	The interrupt request is asserted
SCR[IEF] and SCR[CFF] are set	The interrupt request is asserted
SCR[IER] and SCR[CFR] are cleared for a rising-edge interrupt	The interrupt request is deasserted
SCR[IEF] and SCR[CFF] are cleared for a falling-edge interrupt	The interrupt request is deasserted

25.5 DMA support

Normally, the CMP generates a CPU interrupt if there is a change on the COUT. When DMA support is enabled by setting SCR[DMAEN] and the interrupt is enabled by setting SCR[IER], SCR[IEF], or both, the corresponding change on COUT forces a DMA transfer request rather than a CPU interrupt instead. When the DMA has completed the transfer, it sends a transfer completing indicator that deasserts the DMA transfer request and clears the flag to allow a subsequent change on comparator output to occur and force another DMA request.

The comparator can remain functional in STOP modes.

When DMA support is enabled by setting SCR[DMAEN] and the interrupt is enabled by setting SCR[IER], SCR[IEF], or both, the corresponding change on COUT forces a DMA transfer request to wake up the system from STOP modes. After the data transfer has finished, system will go back to STOP modes. Refer to DMA chapters in the device reference manual for the asynchronous DMA function for details.

25.6 CMP Asynchronous DMA support

The comparator can remain functional in STOP modes.

When DMA support is enabled by setting SCR[DMAEN] and the interrupt is enabled by setting SCR[IER], SCR[IEF], or both, the corresponding change on COUT forces a DMA transfer request to wake up the system from STOP modes. After the data transfer has finished, system will go back to STOP modes. Refer to DMA chapters in the device reference manual for the asynchronous DMA function for details.

25.7 Digital-to-analog converter

The figure found here shows the block diagram of the DAC module.

It contains a 64-tap resistor ladder network and a 64-to-1 multiplexer, which selects an output voltage from one of 64 distinct levels that outputs from DACO. It is controlled through the DAC Control Register (DACCR). Its supply reference source can be selected from two sources V_{in1} and V_{in2}. The module can be powered down or disabled when not in use. When in Disabled mode, DACO is connected to the analog ground.

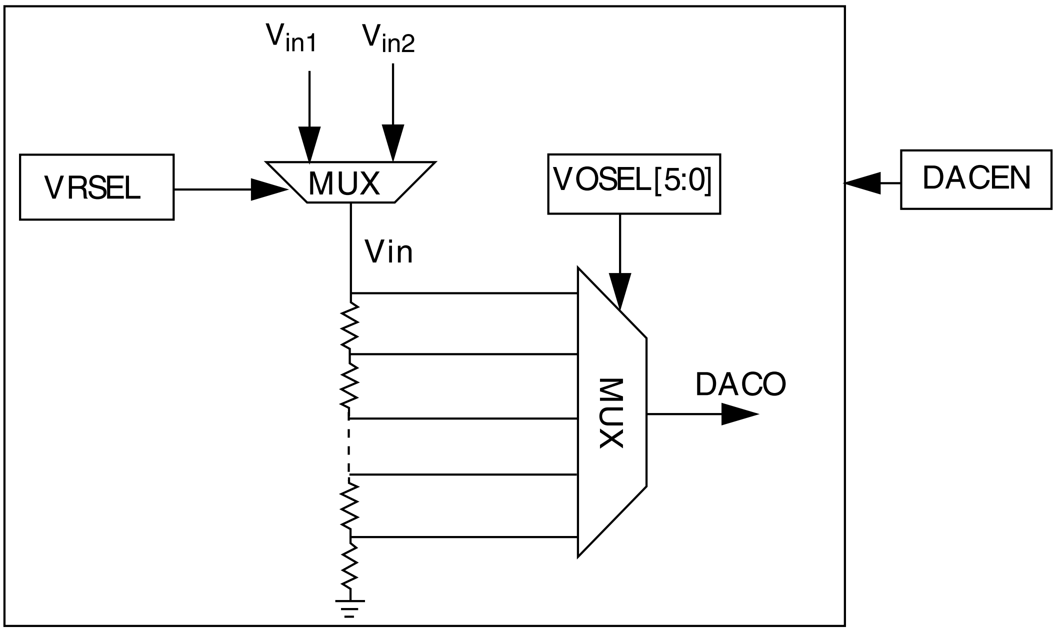


Figure 25-18. 6-bit DAC block diagram

25.8 DAC functional description

This section provides DAC functional description information.

25.8.1 Voltage reference source select

- V_{in1} connects to the primary voltage source as supply reference of 64 tap resistor ladder
- V_{in2} connects to an alternate voltage source

25.9 DAC resets

This module has a single reset input, corresponding to the chip-wide peripheral reset.

25.10 DAC clocks

This module has a single clock input, the bus clock.

25.11 DAC interrupts

This module has no interrupts.

25.12 CMP Trigger Mode

CMP and DAC are configured to CMP Trigger mode when CMP_CR1[TRIGM] is set to 1.

In addition, the CMP must be enabled. If the DAC is to be used as a reference to the CMP, it must also be enabled.

CMP Trigger mode depends on an external timer resource to periodically enable the CMP and 6-bit DAC in order to generate a triggered compare.

Upon setting TRIGM, the CMP and DAC are placed in a standby state until an external timer resource trigger is received.

See the chip configuration chapter for details about the external timer resource.

Chapter 26
12-bit Digital-to-Analog Converter (DAC)

26.1 Introduction

NOTE

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

The 12-bit digital-to-analog converter (DAC) is a low-power, general-purpose DAC. The output of the DAC can be placed on an external pin or set as one of the inputs to the analog comparator, op-amps, or ADC.

26.2 Features

The features of the DAC module include:

- On-chip programmable reference generator output. The voltage output range is from 1⁄4096 V_{in} to V_{in}, and the step is 1⁄4096 V_{in}, where V_{in} is the input voltage.
- V_{in} can be selected from two reference sources
- Static operation in Normal Stop mode
- 2-word data buffer supported with multiple operation modes
- DMA support

26.3 Block diagram

The block diagram of the DAC module is as follows:

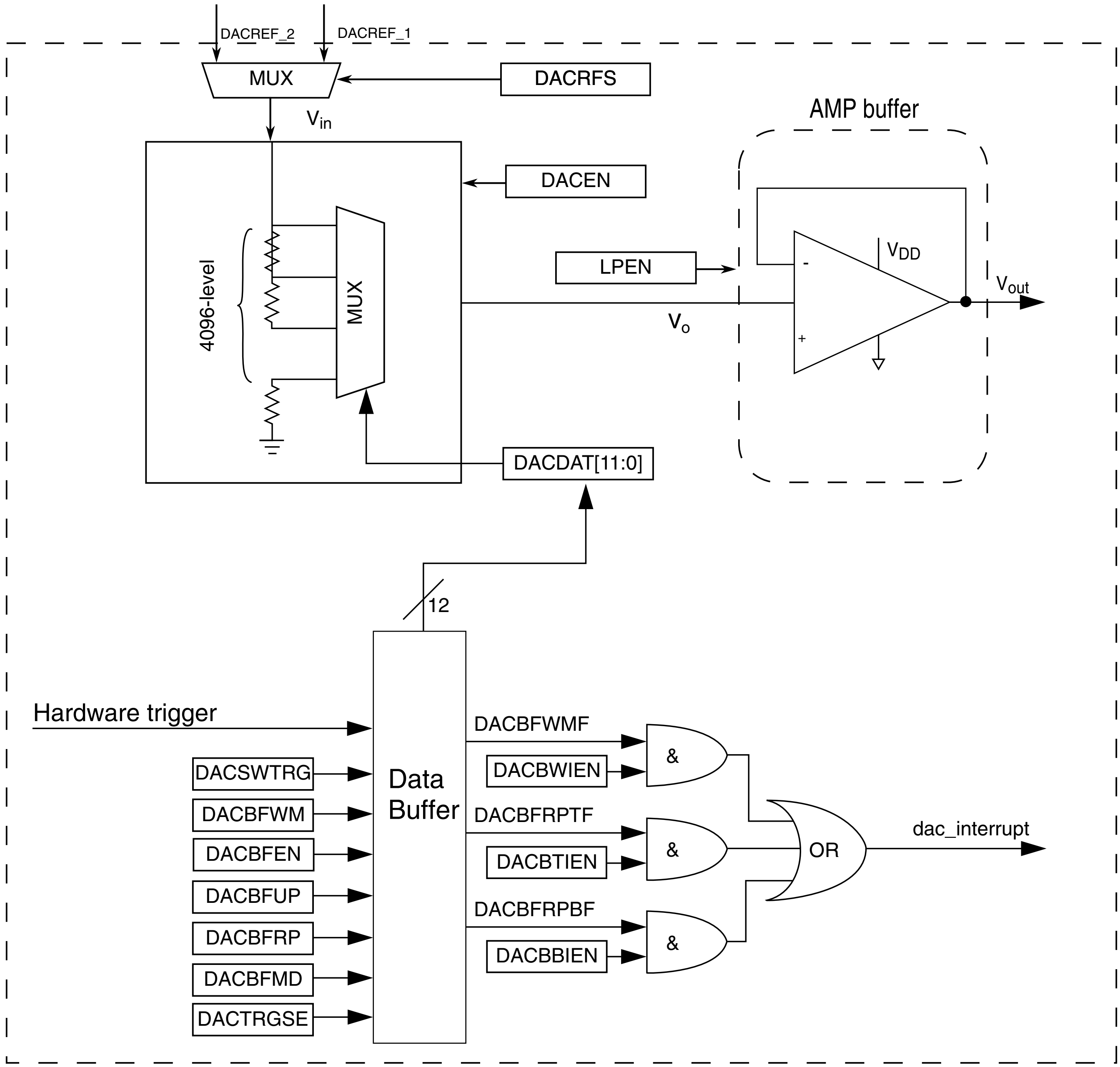


Figure 26-1. DAC block diagram

26.4 Memory map/register definition

The DAC has registers to control analog comparator and programmable voltage divider to perform the digital-to-analog functions.

DAC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_F000	DAC Data Low Register (DAC0_DAT0L)	8	R/W	00h	26.4.1/493
4003_F001	DAC Data High Register (DAC0_DAT0H)	8	R/W	00h	26.4.2/493
4003_F002	DAC Data Low Register (DAC0_DAT1L)	8	R/W	00h	26.4.1/493
4003_F003	DAC Data High Register (DAC0_DAT1H)	8	R/W	00h	26.4.2/493
4003_F020	DAC Status Register (DAC0_SR)	8	R/W	02h	26.4.3/494
4003_F021	DAC Control Register (DAC0_C0)	8	R/W	00h	26.4.4/494
4003_F022	DAC Control Register 1 (DAC0_C1)	8	R/W	00h	26.4.5/496
4003_F023	DAC Control Register 2 (DAC0_C2)	8	R/W	01h	26.4.6/496

26.4.1 DAC Data Low Register (DACx_DAT*n*L)

Address: 4003_F000h base + 0h offset + (2d × i), where i=0d to 1d

Bit	7	6	5	4	3	2	1	0
Read / Write	DATA0							
Reset	0	0	0	0	0	0	0	0

DACx_DAT*n*L field descriptions

Field	Description
7–0 DATA0	When the DAC buffer is not enabled, DATA[11:0] controls the output voltage based on the following formula: $V_{out} = V_{in} * (1 + DACDAT0[11:0])/4096$ When the DAC buffer is enabled, DATA is mapped to the 16-word buffer.

26.4.2 DAC Data High Register (DACx_DAT*n*H)

Address: 4003_F000h base + 1h offset + (2d × i), where i=0d to 1d

Bit	7	6	5	4	3	2	1	0
Read	0				DATA1			
Write								
Reset	0	0	0	0	0	0	0	0

DACx_DAT*n*H field descriptions

Field	Description
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 DATA1	When the DAC Buffer is not enabled, DATA[11:0] controls the output voltage based on the following formula. $V_{out} = V_{in} * (1 + DACDAT0[11:0])/4096$ When the DAC buffer is enabled, DATA[11:0] is mapped to the 16-word buffer.

26.4.3 DAC Status Register (DACx_SR)

If DMA is enabled, the flags can be cleared automatically by DMA when the DMA request is done. Writing 0 to a field clears it whereas writing 1 has no effect. After reset, DACBFRPTF is set and can be cleared by software, if needed. The flags are set only when the data buffer status is changed.

NOTE

Do not use 32/16-bit accesses to this register.

Address: 4003_F000h base + 20h offset = 4003_F020h

Bit	7	6	5	4	3	2	1	0
Read	0						DACBFRPTF	DACBFRPBF
Write								
Reset	0	0	0	0	0	0	1	0

DACx_SR field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DACBFRPTF	DAC Buffer Read Pointer Top Position Flag 0 The DAC buffer read pointer is not zero. 1 The DAC buffer read pointer is zero.
0 DACBFRPBF	DAC Buffer Read Pointer Bottom Position Flag 0 The DAC buffer read pointer is not equal to C2[DACBFUP]. 1 The DAC buffer read pointer is equal to C2[DACBFUP].

26.4.4 DAC Control Register (DACx_C0)

NOTE

Do not use 32- or 16-bit accesses to this register.

Address: 4003_F000h base + 21h offset = 4003_F021h

Bit	7	6	5	4	3	2	1	0
Read	DACEN	DACRFS	DACTRGSEL	0	LPEN	0	DACBTIEN	DACBBIEN
Write				DACSWTRG				
Reset	0	0	0	0	0	0	0	0

DACx_C0 field descriptions

Field	Description
7 DACEN	DAC Enable Starts the Programmable Reference Generator operation. 0 The DAC system is disabled. 1 The DAC system is enabled.
6 DACRFS	DAC Reference Select 0 The DAC selects DACREF_1 as the reference voltage. 1 The DAC selects DACREF_2 as the reference voltage.
5 DACTRGSEL	DAC Trigger Select 0 The DAC hardware trigger is selected. 1 The DAC software trigger is selected.
4 DACSWTRG	DAC Software Trigger Active high. This is a write-only field, which always reads 0. If DAC software trigger is selected and buffer is enabled, writing 1 to this field will advance the buffer read pointer once. 0 The DAC soft trigger is not valid. 1 The DAC soft trigger is valid.
3 LPEN	DAC Low Power Control **NOTE:** See the 12-bit DAC electrical characteristics of the device data sheet for details on the impact of the modes below. 0 High-Power mode 1 Low-Power mode
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DACBTIEN	DAC Buffer Read Pointer Top Flag Interrupt Enable 0 The DAC buffer read pointer top flag interrupt is disabled. 1 The DAC buffer read pointer top flag interrupt is enabled.
0 DACBBIEN	DAC Buffer Read Pointer Bottom Flag Interrupt Enable 0 The DAC buffer read pointer bottom flag interrupt is disabled. 1 The DAC buffer read pointer bottom flag interrupt is enabled.

26.4.5 DAC Control Register 1 (DACx_C1)

NOTE

Do not use 32- or 16-bit accesses to this register.

Address: 4003_F000h base + 22h offset = 4003_F022h

Bit	7	6	5	4	3	2	1	0
Read	DMAEN	0				DACBFMD	0	DACBFEN
Write	DMAEN					DACBFMD		DACBFEN
Reset	0	0	0	0	0	0	0	0

DACx_C1 field descriptions

Field	Description
7 DMAEN	DMA Enable Select 0 DMA is disabled. 1 DMA is enabled. When DMA is enabled, the DMA request will be generated by original interrupts. The interrupts will not be presented on this module at the same time.
6–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 DACBFMD	DAC Buffer Work Mode Select 0 Normal mode 1 One-Time Scan mode
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DACBFEN	DAC Buffer Enable 0 Buffer read pointer is disabled. The converted data is always the first word of the buffer. 1 Buffer read pointer is enabled. The converted data is the word that the read pointer points to. It means converted data can be from any word of the buffer.

26.4.6 DAC Control Register 2 (DACx_C2)

Address: 4003_F000h base + 23h offset = 4003_F023h

Bit	7	6	5	4	3	2	1	0
Read	0			DACBFRP	0			DACBFUP
Write				DACBFRP				DACBFUP
Reset	0	0	0	0	0	0	0	1

DACx_C2 field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 DACBFRP	DAC Buffer Read Pointer Keeps the current value of the buffer read pointer.
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DACBFUP	DAC Buffer Upper Limit Selects the upper limit of the DAC buffer. The buffer read pointer cannot exceed it.

26.5 Functional description

The 12-bit DAC module can select one of the two reference inputs—DACREF_1 and DACREF_2 as the DAC reference voltage, V_{in} by C0[DACRFS]. See the module introduction for information on the source for DACREF_1 and DACREF_2.

When the DAC is enabled, it converts the data in DACDAT0[11:0] or the data from the DAC data buffer to a stepped analog output voltage. The output voltage range is from V_{in} to $V_{in}/4096$, and the step is $V_{in}/4096$.

26.5.1 DAC data buffer operation

When the DAC is enabled and the buffer is not enabled, the DAC module always converts the data in DAT0 to analog output voltage.

When both the DAC and the buffer are enabled, the DAC converts the data in the data buffer to analog output voltage. The data buffer read pointer advances to the next word whenever any hardware or software trigger event occurs. Refer to Introduction for the hardware trigger connection.

The data buffer can be configured to operate in Normal mode, Swing mode, One-Time Scan mode or FIFO mode. When the buffer operation is switched from one mode to another, the read pointer does not change. The read pointer can be set to any value between 0 and C2[DACBFUP] by writing C2[DACBFRP].

26.5.1.1 DAC data buffer interrupts

There are several interrupts and associated flags that can be configured for the DAC buffer. SR[DACBFRPBF] is set when the DAC buffer read pointer reaches the DAC buffer upper limit, that is, C2[DACBFRP] = C2[DACBFUP]. SR[DACBFRPTF] is set when the DAC read pointer is equal to the start position, 0.

26.5.1.2 Modes of DAC data buffer operation

The following table describes the different modes of data buffer operation for the DAC module.

Table 26-23. Modes of DAC data buffer operation

Modes	Description
Buffer Normal mode	This is the default mode. The buffer works as a circular buffer. The read pointer increases by one, every time the trigger occurs. When the read pointer reaches the upper limit, it goes to 0 directly in the next trigger event.
Buffer One-time Scan mode	The read pointer increases by 1 every time the trigger occurs. When it reaches the upper limit, it stops there. If read pointer is reset to the address other than the upper limit, it will increase to the upper address and stop there again. **NOTE:** If the software set the read pointer to the upper limit, the read pointer will not advance in this mode.

26.5.2 DMA operation

When DMA is enabled, DMA requests are generated instead of interrupt requests. The DMA Done signal clears the DMA request.

The status register flags are still set and are cleared automatically when the DMA completes.

26.5.3 Resets

During reset, the DAC is configured in the default mode and is disabled.

26.5.4 Low-Power mode operation

The following table shows the wait mode and the stop mode operation of the DAC module.

Table 26-24. Modes of operation

Modes of operation	Description
Wait mode	The DAC will operate normally, if enabled.
Stop mode	If enabled, the DAC module continues to operate in Normal Stop mode and the output voltage will hold the value before stop. In low-power stop modes, the DAC is fully shut down.

NOTE

The assignment of module modes to core modes is chip-specific. For module-to-core mode assignments, see the chapter that describes how modules are configured.

Chapter 27
Timer/PWM Module (TPM)

27.1 Introduction

The TPM (Timer/PWM Module) is a 2- to 8-channel timer which supports input capture, output compare, and the generation of PWM signals to control electric motor and power management applications.

The counter, compare and capture registers are clocked by an asynchronous clock that can remain enabled in low power modes. An example of using the TPM with the asynchronous DMA is described in AN4631:Using the Asynchronous DMA features of the Kinetis L Series .

27.1.1 TPM Philosophy

The TPM is built upon a very simple timer (HCS08 Timer PWM Module – TPM) used for many years on Freescale's 8-bit microcontrollers. The TPM extends the functionality to support operation in low power modes by clocking the counter, compare and capture registers from an asynchronous clock that can remain functional in low power modes.

27.1.2 Features

The TPM features include:

- TPM clock mode is selectable
  - Can increment on every edge of the asynchronous counter clock
  - Can increment on rising edge of an external clock input synchronized to the asynchronous counter clock
- Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128

- TPM includes a 16-bit counter
    - It can be a free-running counter or modulo counter
    - The counting can be up or up-down
- Includes 6 channels that 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
    - In output compare mode the output signal can be set, cleared, pulsed, or toggled on match
    - All channels can be configured for center-aligned PWM mode
- Support the generation of an interrupt and/or DMA request per channel
- Support the generation of an interrupt and/or DMA request when the counter overflows
- Support selectable trigger input to optionally reset or cause the counter to start incrementing.
    - The counter can also optionally stop incrementing on counter overflow
- Support the generation of hardware triggers when the counter overflows and per channel

27.1.3 Modes of operation

During debug mode, the TPM can can be configured to temporarily pause all counting until the core returns to normal user operating mode or to operate normally. When the counter is paused, trigger inputs and input capture events are ignored.

During doze mode, the TPM can be configured to operate normally or to pause all counting for the duration of doze mode. When the counter is paused, trigger inputs and input capture events are ignored.

During stop mode, the TPM counter clock can remain functional and the TPM can generate an asynchronous interrupt to exit the MCU from stop mode.

27.1.4 Block diagram

The TPM uses one input/output (I/O) pin per channel, CHn (TPM channel (n)) where n is the channel number.

The following figure shows the TPM structure. The central component of the TPM is the 16-bit counter with programmable final value and its counting can be up or up-down.

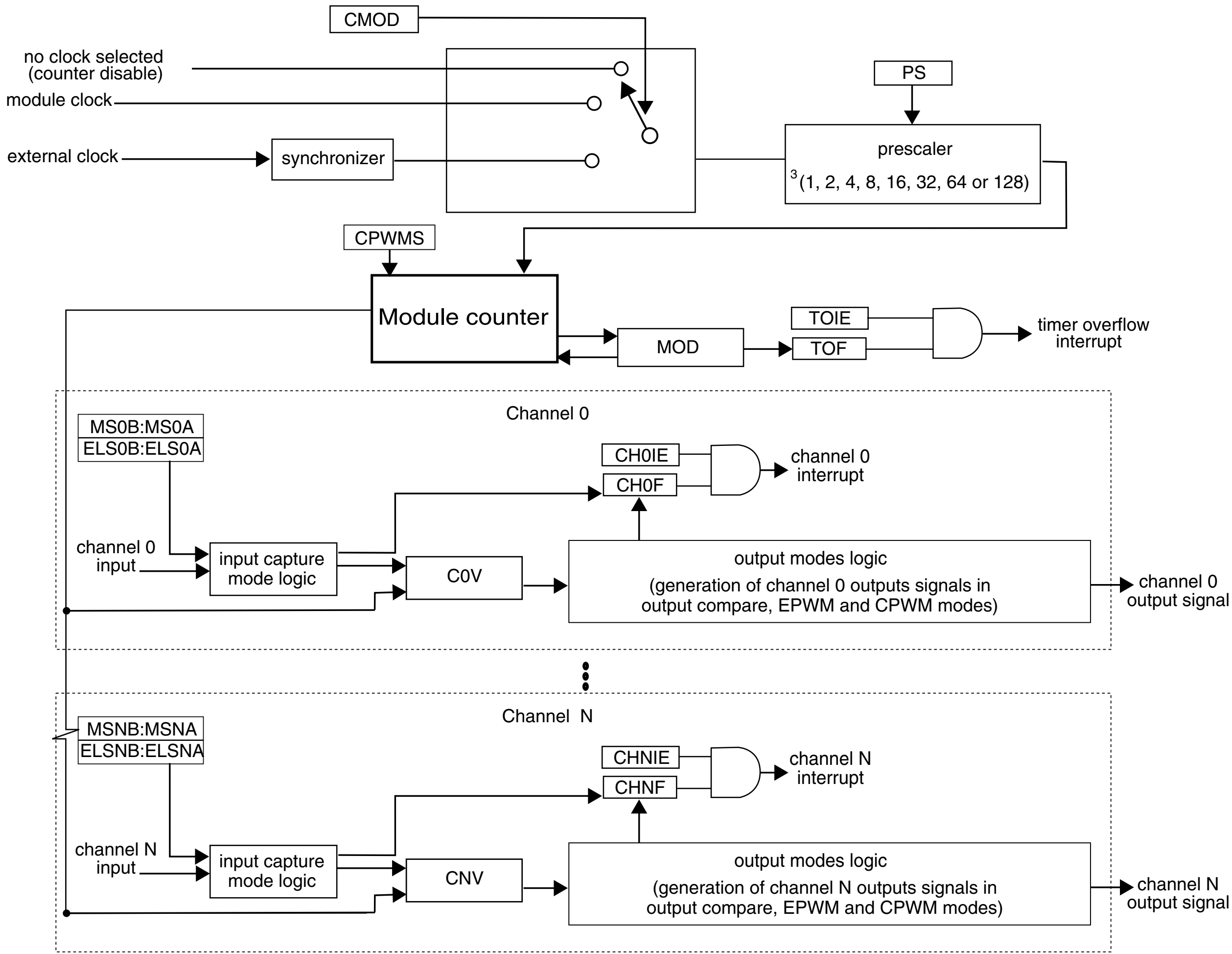


Figure 27-1. TPM block diagram

27.2 TPM Signal Descriptions

Table 27-1 shows the user-accessible signals for the TPM.

Table 27-1. TPM signal descriptions

Signal	Description	I/O
TPM_EXTCLK	External clock. TPM external clock can be selected to increment the TPM counter on every rising edge synchronized to the counter clock.	I

Table continues on the next page...

Table 27-1. TPM signal descriptions (continued)

Signal	Description	I/O
TPM_CHn	TPM channel (n = 5 to 0)	I/O

27.2.1 TPM_EXTCLK — TPM External Clock

The rising edge of the external input signal is used to increment the TPM counter if selected by CMOD[1:0] bits in the SC register. This input signal must be less than half of the TPM counter clock frequency. The TPM counter prescaler selection and settings are also used when an external input is selected.

27.2.2 TPM_CHn — TPM Channel (n) I/O Pin

Each TPM channel can be configured to operate either as input or output. The direction associated with each channel, input or output, is selected according to the mode assigned for that channel.

27.3 Memory Map and Register Definition

This section provides a detailed description of all TPM registers.

Attempting to access a reserved register location in the TPM memory map will generate a bus error.

TPM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8000	Status and Control (TPM0_SC)	32	R/W	0000_0000h	27.3.1/506
4003_8004	Counter (TPM0_CNT)	32	R/W	0000_0000h	27.3.2/508
4003_8008	Modulo (TPM0_MOD)	32	R/W	0000_FFFFh	27.3.3/508
4003_800C	Channel (n) Status and Control (TPM0_C0SC)	32	R/W	0000_0000h	27.3.4/509
4003_8010	Channel (n) Value (TPM0_C0V)	32	R/W	0000_0000h	27.3.5/511
4003_8014	Channel (n) Status and Control (TPM0_C1SC)	32	R/W	0000_0000h	27.3.4/509
4003_8018	Channel (n) Value (TPM0_C1V)	32	R/W	0000_0000h	27.3.5/511
4003_801C	Channel (n) Status and Control (TPM0_C2SC)	32	R/W	0000_0000h	27.3.4/509
4003_8020	Channel (n) Value (TPM0_C2V)	32	R/W	0000_0000h	27.3.5/511

Table continues on the next page...

TPM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8024	Channel (n) Status and Control (TPM0_C3SC)	32	R/W	0000_0000h	27.3.4/509
4003_8028	Channel (n) Value (TPM0_C3V)	32	R/W	0000_0000h	27.3.5/511
4003_802C	Channel (n) Status and Control (TPM0_C4SC)	32	R/W	0000_0000h	27.3.4/509
4003_8030	Channel (n) Value (TPM0_C4V)	32	R/W	0000_0000h	27.3.5/511
4003_8034	Channel (n) Status and Control (TPM0_C5SC)	32	R/W	0000_0000h	27.3.4/509
4003_8038	Channel (n) Value (TPM0_C5V)	32	R/W	0000_0000h	27.3.5/511
4003_8050	Capture and Compare Status (TPM0_STATUS)	32	R/W	0000_0000h	27.3.6/512
4003_8084	Configuration (TPM0_CONF)	32	R/W	0000_0000h	27.3.7/514
4003_9000	Status and Control (TPM1_SC)	32	R/W	0000_0000h	27.3.1/506
4003_9004	Counter (TPM1_CNT)	32	R/W	0000_0000h	27.3.2/508
4003_9008	Modulo (TPM1_MOD)	32	R/W	0000_FFFFh	27.3.3/508
4003_900C	Channel (n) Status and Control (TPM1_C0SC)	32	R/W	0000_0000h	27.3.4/509
4003_9010	Channel (n) Value (TPM1_C0V)	32	R/W	0000_0000h	27.3.5/511
4003_9014	Channel (n) Status and Control (TPM1_C1SC)	32	R/W	0000_0000h	27.3.4/509
4003_9018	Channel (n) Value (TPM1_C1V)	32	R/W	0000_0000h	27.3.5/511
4003_901C	Channel (n) Status and Control (TPM1_C2SC)	32	R/W	0000_0000h	27.3.4/509
4003_9020	Channel (n) Value (TPM1_C2V)	32	R/W	0000_0000h	27.3.5/511
4003_9024	Channel (n) Status and Control (TPM1_C3SC)	32	R/W	0000_0000h	27.3.4/509
4003_9028	Channel (n) Value (TPM1_C3V)	32	R/W	0000_0000h	27.3.5/511
4003_902C	Channel (n) Status and Control (TPM1_C4SC)	32	R/W	0000_0000h	27.3.4/509
4003_9030	Channel (n) Value (TPM1_C4V)	32	R/W	0000_0000h	27.3.5/511
4003_9034	Channel (n) Status and Control (TPM1_C5SC)	32	R/W	0000_0000h	27.3.4/509
4003_9038	Channel (n) Value (TPM1_C5V)	32	R/W	0000_0000h	27.3.5/511
4003_9050	Capture and Compare Status (TPM1_STATUS)	32	R/W	0000_0000h	27.3.6/512
4003_9084	Configuration (TPM1_CONF)	32	R/W	0000_0000h	27.3.7/514
4003_A000	Status and Control (TPM2_SC)	32	R/W	0000_0000h	27.3.1/506
4003_A004	Counter (TPM2_CNT)	32	R/W	0000_0000h	27.3.2/508
4003_A008	Modulo (TPM2_MOD)	32	R/W	0000_FFFFh	27.3.3/508
4003_A00C	Channel (n) Status and Control (TPM2_C0SC)	32	R/W	0000_0000h	27.3.4/509
4003_A010	Channel (n) Value (TPM2_C0V)	32	R/W	0000_0000h	27.3.5/511
4003_A014	Channel (n) Status and Control (TPM2_C1SC)	32	R/W	0000_0000h	27.3.4/509
4003_A018	Channel (n) Value (TPM2_C1V)	32	R/W	0000_0000h	27.3.5/511
4003_A01C	Channel (n) Status and Control (TPM2_C2SC)	32	R/W	0000_0000h	27.3.4/509
4003_A020	Channel (n) Value (TPM2_C2V)	32	R/W	0000_0000h	27.3.5/511
4003_A024	Channel (n) Status and Control (TPM2_C3SC)	32	R/W	0000_0000h	27.3.4/509
4003_A028	Channel (n) Value (TPM2_C3V)	32	R/W	0000_0000h	27.3.5/511
4003_A02C	Channel (n) Status and Control (TPM2_C4SC)	32	R/W	0000_0000h	27.3.4/509
4003_A030	Channel (n) Value (TPM2_C4V)	32	R/W	0000_0000h	27.3.5/511

Table continues on the next page...

TPM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_A034	Channel (n) Status and Control (TPM2_C5SC)	32	R/W	0000_0000h	27.3.4/509
4003_A038	Channel (n) Value (TPM2_C5V)	32	R/W	0000_0000h	27.3.5/511
4003_A050	Capture and Compare Status (TPM2_STATUS)	32	R/W	0000_0000h	27.3.6/512
4003_A084	Configuration (TPM2_CONF)	32	R/W	0000_0000h	27.3.7/514

27.3.1 Status and Control (TPMx_SC)

SC contains the overflow status flag and control bits used to configure the interrupt enable, module configuration and prescaler factor. These controls relate to all channels within this module.

Address: Base address + 0h offset

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							DMA	TOF	TOIE	CPWMS	CMOD		PS		
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_SC field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 DMA	DMA Enable Enables DMA transfers for the overflow flag.

Table continues on the next page...

TPMx_SC field descriptions (continued)

Field	Description
	0 Disables DMA transfers. 1 Enables DMA transfers.
7 TOF	Timer Overflow Flag Set by hardware when the TPM counter equals the value in the MOD register and increments. Writing a 1 to TOF clears it. Writing a 0 to TOF has no effect. If another LPTPM overflow occurs between the flag setting and the flag clearing, the write operation has no effect; therefore, TOF remains set indicating another overflow has occurred. In this case a TOF interrupt request is not lost due to a delay in clearing the previous TOF. 0 TPM counter has not overflowed. 1 TPM counter has overflowed.
6 TOIE	Timer Overflow Interrupt Enable Enables TPM overflow interrupts. 0 Disable TOF interrupts. Use software polling or DMA request. 1 Enable TOF interrupts. An interrupt is generated when TOF equals one.
5 CPWMS	Center-Aligned PWM Select Selects CPWM mode. This mode configures the TPM to operate in up-down counting mode. This field is write protected. It can be written only when the counter is disabled. 0 TPM counter operates in up counting mode. 1 TPM counter operates in up-down counting mode.
4–3 CMOD	Clock Mode Selection Selects the TPM counter clock modes. When disabling the counter, this field remain set until acknolwedged in the TPM clock domain. 00 TPM counter is disabled 01 TPM counter increments on every TPM counter clock 10 TPM counter increments on rising edge of TPM_EXTCLK synchronized to the TPM counter clock 11 Reserved
2–0 PS	Prescale Factor Selection Selects one of 8 division factors for the clock mode selected by CMOD. This field is write protected. It can be written only when the counter is disabled. 000 Divide by 1 001 Divide by 2 010 Divide by 4 011 Divide by 8 100 Divide by 16 101 Divide by 32 110 Divide by 64 111 Divide by 128

27.3.2 Counter (TPMx_CNT)

The CNT register contains the TPM counter value.

Reset clears the CNT register. Writing any value to COUNT also clears the counter.

When debug is active, the TPM counter does not increment unless configured otherwise.

Reading the CNT register adds two wait states to the register access due to synchronization delays.

Address: Base address + 4h offset

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

TPMx_CNT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 COUNT	Counter value

27.3.3 Modulo (TPMx_MOD)

The Modulo register contains the modulo value for the TPM counter. When the TPM counter reaches the modulo value and increments, the overflow flag (TOF) is set and the next value of TPM counter depends on the selected counting method (see Counter).

Writing to the MOD register latches the value into a buffer. The MOD register is updated with the value of its write buffer according to MOD Register Update . Additional writes to the MOD write buffer are ignored until the register has been updated.

It is recommended to initialize the TPM counter (write to CNT) before writing to the MOD register to avoid confusion about when the first counter overflow will occur.

Address: Base address + 8h offset

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

TPMx_MOD field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 MOD	Modulo value When writing this field, all bytes must be written at the same time.

27.3.4 Channel (n) Status and Control (TPMx_C*n*SC)

CnSC contains the channel-interrupt-status flag and control bits used to configure the interrupt enable, channel configuration, and pin function. When switching from one channel mode to a different channel mode, the channel must first be disabled and this must be acknowledged in the TPM counter clock domain.

Table 27-34. Mode, Edge, and Level Selection

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	00	00	None	Channel disabled
X	01	00	Software compare	Pin not used for TPM
0	00	01	Input capture	Capture on Rising Edge Only
		10		Capture on Falling Edge Only
		11		Capture on Rising or Falling Edge
	01	01	Output compare	Toggle Output on match
		10		Clear Output on match
		11		Set Output on match
	10	10	Edge-aligned PWM	High-true pulses (clear Output on match, set Output on reload)
		X1		Low-true pulses (set Output on match, clear Output on reload)
	11	10	Output compare	Pulse Output low on match
		01		Pulse Output high on match

Table continues on the next page...

Table 27-34. Mode, Edge, and Level Selection (continued)

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
1	10	10	Center-aligned PWM	High-true pulses (clear Output on match-up, set Output on match-down)
		01		Low-true pulses (set Output on match-up, clear Output on match-down)

Address: Base address + Ch offset + (8d × i), where i=0d to 5d

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CHF	CHIE	MSB	MSA	ELSB	ELSA	0	DMA
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_CnSC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CHF	Channel Flag Set by hardware when an event occurs on the channel. CHF is cleared by writing a 1 to the CHF bit. Writing a 0 to CHF has no effect. If another event occurs between the CHF sets and the write operation, the write operation has no effect; therefore, CHF remains set indicating another event has occurred. In this case a CHF interrupt request is not lost due to the delay in clearing the previous CHF. 0 No channel event has occurred. 1 A channel event has occurred.
6 CHIE	Channel Interrupt Enable Enables channel interrupts. 0 Disable channel interrupts. 1 Enable channel interrupts.
5 MSB	Channel Mode Select Used for further selections in the channel logic. Its functionality is dependent on the channel mode. When a channel is disabled, this field will not change state until acknowledged in the TPM counter clock domain.
4 MSA	Channel Mode Select

Table continues on the next page...

TPMx_CnSC field descriptions (continued)

Field	Description
	Used for further selections in the channel logic. Its functionality is dependent on the channel mode. When a channel is disabled, this field will not change state until acknowledged in the TPM counter clock domain.
3 ELSB	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. When a channel is disabled, this field will not change state until acknowledged in the TPM counter clock domain.
2 ELSA	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. When a channel is disabled, this field will not change state until acknowledged in the TPM counter clock domain.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DMA	DMA Enable Enables DMA transfers for the channel. 0 Disable DMA transfers. 1 Enable DMA transfers.

27.3.5 Channel (n) Value (TPMx_CnV)

These registers contain the captured TPM counter value for the input modes or the match value for the output modes.

In input capture mode, any write to a CnV register is ignored.

In compare modes, writing to a CnV register latches the value into a buffer. A CnV register is updated with the value of its write buffer according to CnV Register Update . Additional writes to the CnV write buffer are ignored until the register has been updated.

Address: Base address + 10h offset + (8d × i), where i=0d to 5d

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

TPMx_CnV field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 VAL	Channel Value Captured TPM counter value of the input modes or the match value for the output modes. When writing this field, all bytes must be written at the same time.

27.3.6 Capture and Compare Status (TPMx_STATUS)

The STATUS register contains a copy of the status flag, CnSC[CHnF] for each TPM channel, as well as SC[TOF], for software convenience.

Each CHnF bit in STATUS is a mirror of CHnF bit in CnSC. All CHnF bits can be checked using only one read of STATUS. All CHnF bits can be cleared by writing all ones to STATUS.

Hardware sets the individual channel flags when an event occurs on the channel. Writing a 1 to CHF clears it. Writing a 0 to CHF has no effect.

If another event occurs between the flag setting and the write operation, the write operation has no effect; therefore, CHF remains set indicating another event has occurred. In this case a CHF interrupt request is not lost due to the clearing sequence for a previous CHF.

Address: Base address + 50h offset

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							TOF	0		CH5F	CH4F	CH3F	CH2F	CH1F	CH0F
W								w1c			w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPMx_STATUS field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 TOF	Timer Overflow Flag See register description

Table continues on the next page...

TPMx_STATUS field descriptions (continued)

Field	Description
	0 TPM counter has not overflowed. 1 TPM counter has overflowed.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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. 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.

27.3.7 Configuration (TPMx_CONF)

This register selects the behavior in debug and wait modes and the use of an external global time base.

Address: Base address + 84h offset

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

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

TPMx_CONF field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 TRGSEL	Trigger Select Selects the input trigger to use for starting the counter and/or reloading the counter. This field should only be changed when the TPM counter is disabled. See Chip configuration section for available options.
23–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 CROT	Counter Reload On Trigger When set, the TPM counter will reload with 0 (and initialize PWM outputs to their default value) when a rising edge is detected on the selected trigger input. The trigger input is ignored if the TPM counter is paused during debug mode or doze mode. This field should only be changed when the TPM counter is disabled. 0 Counter is not reloaded due to a rising edge on the selected input trigger 1 Counter is reloaded when a rising edge is detected on the selected input trigger
17 CSOO	Counter Stop On Overflow When set, the TPM counter will stop incrementing once the counter equals the MOD value and incremented (this also sets the TOF). Reloading the counter with 0 due to writing to the counter register or due to a trigger input does not cause the counter to stop incrementing. Once the counter has stopped incrementing, the counter will not start incrementing unless it is disabled and then enabled again, or a rising edge on the selected trigger input is detected when CSOT set.

Table continues on the next page...

TPMx_CONF field descriptions (continued)

Field	Description
	This field should only be changed when the TPM counter is disabled. 0 TPM counter continues incrementing or decrementing after overflow 1 TPM counter stops incrementing or decrementing after overflow.
16 CSOT	Counter Start on Trigger When set, the TPM counter will not start incrementing after it is enabled until a rising edge on the selected trigger input is detected. If the TPM counter is stopped due to an overflow, a rising edge on the selected trigger input will also cause the TPM counter to start incrementing again. The trigger input is ignored if the TPM counter is paused during debug mode or doze mode. This field should only be changed when the TPM counter is disabled. 0 TPM counter starts to increment immediately, once it is enabled. 1 TPM counter only starts to increment when it a rising edge on the selected input trigger is detected, after it has been enabled or after it has stopped due to overflow.
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 GTBEEN	Global time base enable Configures the TPM to use an externally generated global time base counter. When an externally generated timebase is used, the internal TPM counter is not used by the channels but can be used to generate a periodic interruptor DMA request using the Modulo register and timer overflow flag. 0 All channels use the internally generated TPM counter as their timebase 1 All channels use an externally generated global timebase as their timebase
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 DBGMODE	Debug Mode Configures the TPM behavior in debug mode. All other configurations are reserved. 00 TPM counter is paused and does not increment during debug mode. Trigger inputs and input capture events are also ignored. 11 TPM counter continues in debug mode.
5 DOZEEN	Doze Enable Configures the TPM behavior in wait mode. 0 Internal TPM counter continues in Doze mode. 1 Internal TPM counter is paused and does not increment during Doze mode. Trigger inputs and input capture events are also ignored.
4–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

27.4 Functional description

The following sections describe the TPM features.

27.4.1 Clock domains

The TPM module supports two clock domains.

The bus clock domain is used by the register interface and for synchronizing interrupts and DMA requests.

The TPM counter clock domain is used to clock the counter and prescaler along with the output compare and input capture logic. The TPM counter clock is considered asynchronous to the bus clock, can be a higher or lower frequency than the bus clock and can remain operational in Stop mode. Multiple TPM instances are all clocked by the same TPM counter clock in support of the external timebase feature.

27.4.1.1 Counter Clock Mode

The CMOD[1:0] bits in the SC register either disable the TPM counter or select one of two possible clock modes for the TPM counter. After any reset, CMOD[1:0] = 0:0 so the TPM counter is disabled.

The CMOD[1:0] bits may be read or written at any time. Disabling the TPM counter by writing zero to the CMOD[1:0] bits does not affect the TPM counter value or other registers, but must be acknowledged by the TPM counter clock domain before they read as zero.

The external clock input passes through a synchronizer clocked by the TPM counter clock to assure that counter transitions are properly aligned to counter clock transitions. Therefore, to meet Nyquist criteria considering also jitter, the frequency of the external clock source must be less than half of the counter clock frequency.

27.4.2 Prescaler

The selected counter clock source passes through a prescaler that is a 7-bit counter.

The value of the prescaler is selected by the PS[2:0] bits. The following figure shows an example of the prescaler counter and TPM counter.

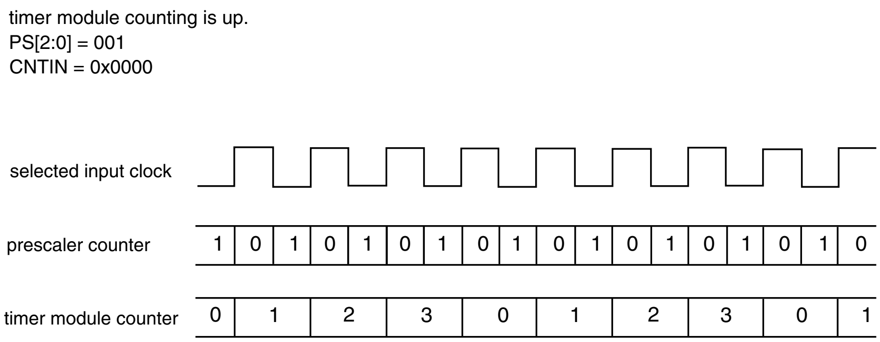


Figure 27-78. Example of the Prescaler Counter

27.4.3 Counter

The TPM has a 16-bit counter that is used by the channels either for input or output modes.

The counter updates from the selected clock divided by the prescaler.

The TPM counter has these modes of operation:

- up counting (see Up counting)
- up-down counting (see Up-down counting)

27.4.3.1 Up counting

Up counting is selected when SC[CPWMS] = 0.

The value of zero is loaded into the TPM counter, and the counter increments until the value of MOD is reached, at which point the counter is reloaded with zero.

The TPM period when using up counting is (MOD + 0x0001) × period of the TPM counter clock.

The TOF bit is set when the TPM counter changes from MOD to zero.

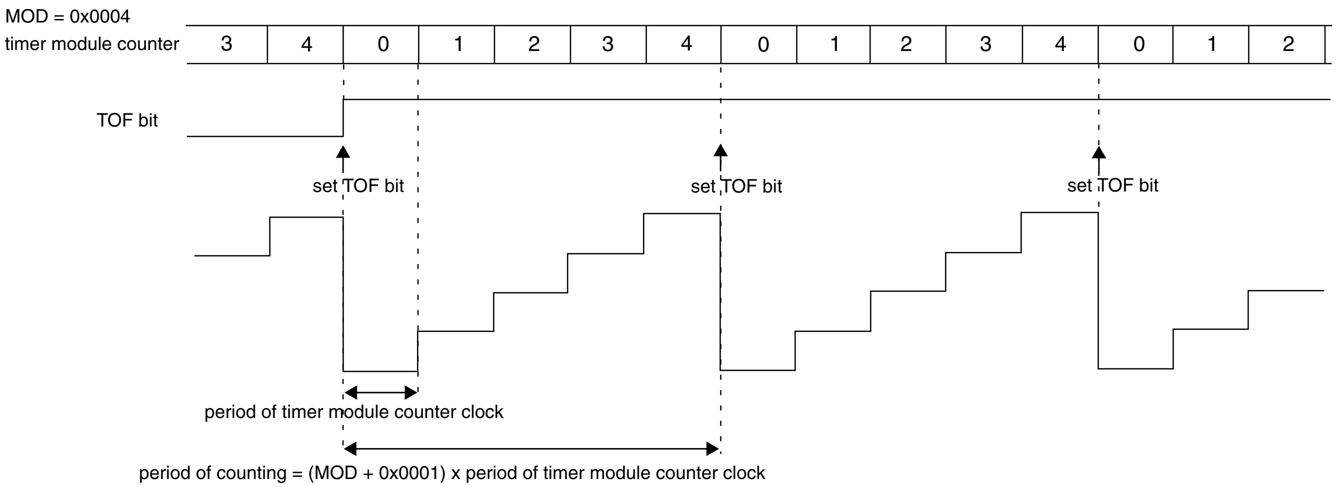


Figure 27-79. Example of TPM Up Counting

Note

- MOD = 0000 is a redundant condition. In this case, the TPM counter is always equal to MOD and the TOF bit is set in each rising edge of the TPM counter clock.

27.4.3.2 Up-down counting

Up-down counting is selected when SC[CPWMS] = 1. When configured for up-down counting, configuring CONF[MOD] to less than 2 is not supported.

The value of 0 is loaded into the TPM counter, and the counter increments until the value of MOD is reached, at which point the counter is decremented until it returns to zero and the up-down counting restarts.

The TPM period when using up-down counting is 2 × MOD × period of the TPM counter clock.

The TOF bit is set when the TPM counter changes from MOD to (MOD – 1).

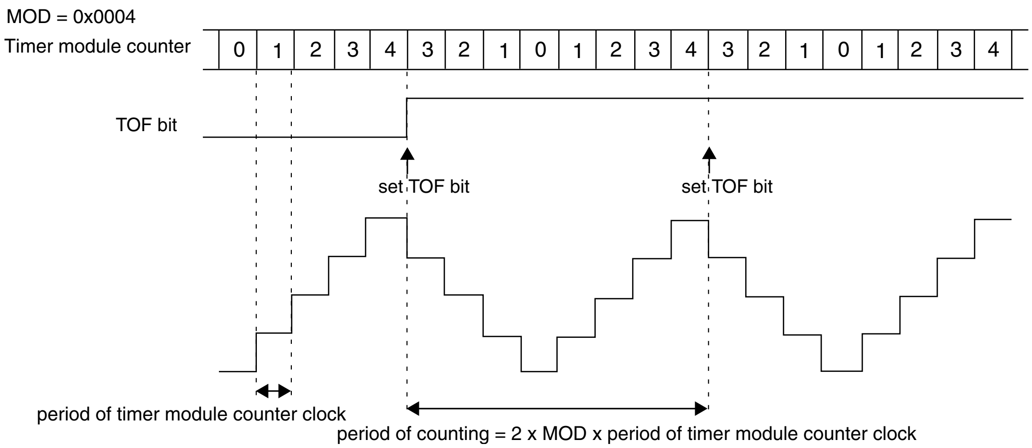


Figure 27-80. Example of up-down counting

27.4.3.3 Counter Reset

Any write to CNT resets the TPM counter and the channel outputs to their initial values (except for channels in output compare mode).

27.4.3.4 Global time base (GTB)

The global time base (GTB) is a TPM function that allows multiple TPM modules to share the same timebase. When the global time base is enabled (CONF[GTBEEN] = 1), the local TPM channels use the counter value, counter enable and overflow indication from the TPM generating the global time base. If the local TPM counter is not generating the global time base, then it can be used as an independent counter or pulse accumulator.

27.4.3.5 Counter trigger

The TPM counter can be configured to start, stop or reset in response to a hardware trigger input. The trigger input is synchronized to the asynchronous counter clock, so there is a 3 counter clock delay between the trigger assertion and the counter responding.

- When (CSOT = 1), the counter will not start incrementing until a rising edge is detected on the trigger input.
- When (CSOO= 1), the counter will stop incrementing whenever the TOF flag is set. The counter does not increment again unless it is disabled, or if CSOT = 1 and a rising edge is detected on the trigger input.
- When (CROT= 1), the counter will reset to zero as if an overflow occurred whenever a rising edge is detected on the trigger input.

27.4.4 Input Capture Mode

The input capture mode is selected when (CPWMS = 0), (MSnB:MSnA = 0:0), and (ELSnB:ELSnA ≠ 0:0).

When a selected edge occurs on the channel input, the current value of the TPM counter is captured into the CnV register, at the same time the CHnF bit is set and the channel interrupt is generated if enabled by CHnIE = 1 (see the following figure).

When a channel is configured for input capture, the TPM_CHn pin is an edge-sensitive input. ELSnB:ELSnA control bits determine which edge, falling or rising, triggers input-capture event. Note that the maximum frequency for the channel input signal to be detected correctly is counter clock divided by 4, which is required to meet Nyquist criteria for signal sampling.

Writes to the CnV register are ignored in input capture mode.

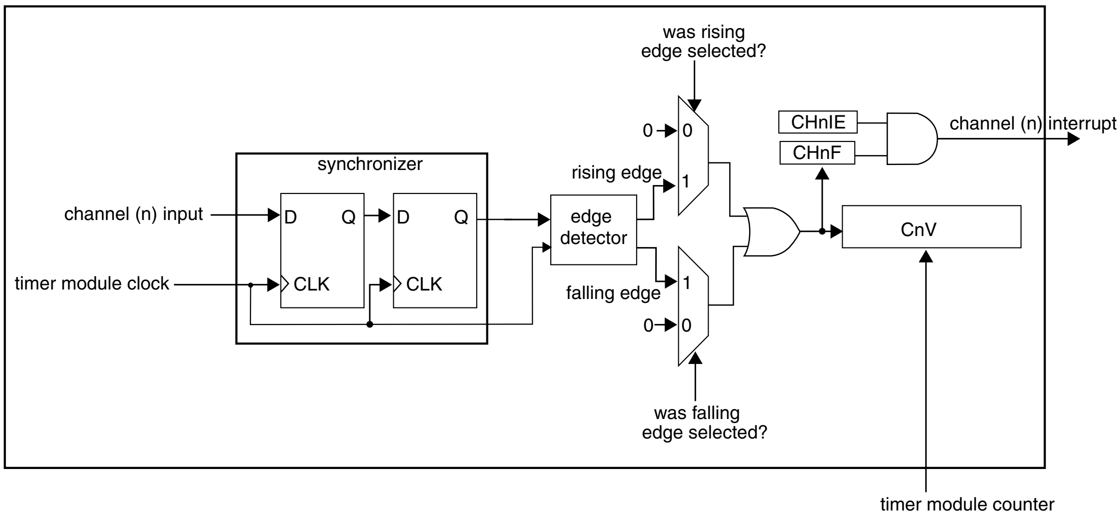


Figure 27-81. Input capture mode

The CHnF bit is set on the third rising edge of the counter clock after a valid edge occurs on the channel input.

27.4.5 Output Compare Mode

The output compare mode is selected when (CPWMS = 0), and (MSnB:MSnA = X:1).

In output compare mode, the TPM can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter matches the value in the CnV register of an output compare channel, the channel (n) output can be set, cleared or toggled if MSnB is clear. If MSnB is set then the channel (n) output is pulsed high or low for as long as the counter matches the value in the CnV register.

When a channel is initially configured to output compare mode, the channel output updates with its negated value (logic 0 for set/toggle/pulse high and logic one for clear/ pulse low).

The CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (TPM counter = CnV).

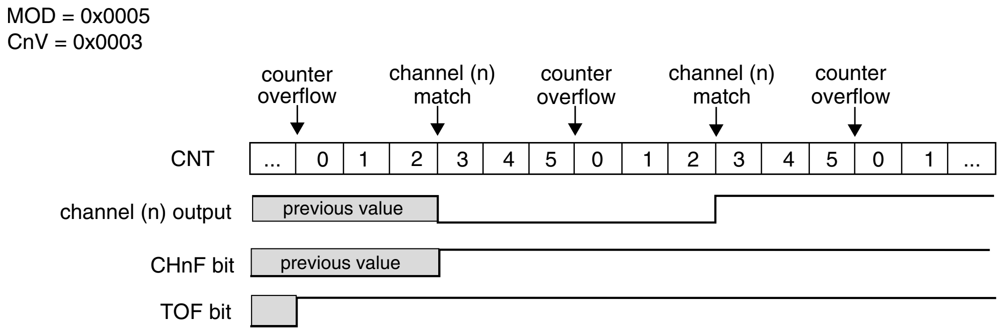


Figure 27-82. Example of the output compare mode when the match toggles the channel output

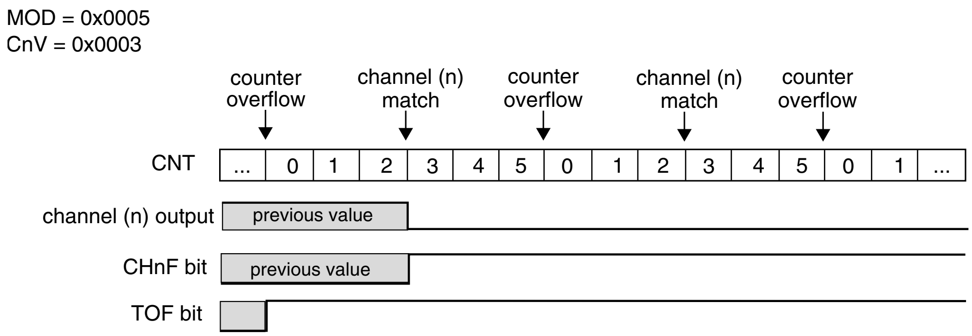


Figure 27-83. Example of the output compare mode when the match clears the channel output

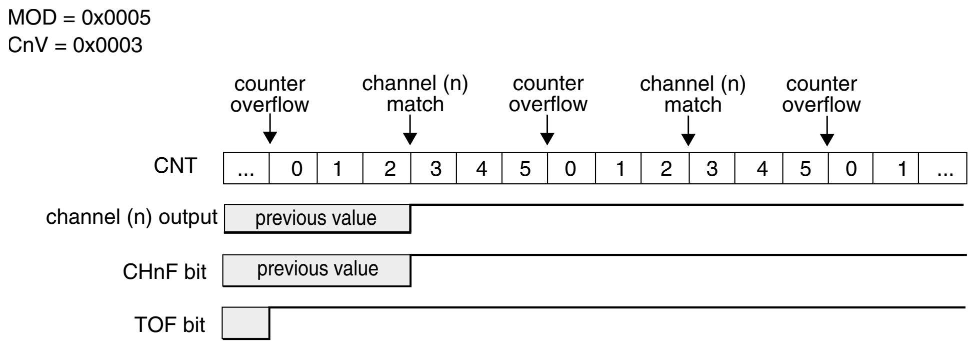


Figure 27-84. Example of the output compare mode when the match sets the channel output

It is possible to use the output compare mode with (ELSnB:ELSnA = 0:0). In this case, when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1), however the channel (n) output is not modified and controlled by TPM.

27.4.6 Edge-Aligned PWM (EPWM) Mode

The edge-aligned mode is selected when (CPWMS = 0), and (MSnB:MSnA = 1:0).

The EPWM period is determined by (MOD + 0x0001) and the pulse width (duty cycle) is determined by CnV.

The CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (TPM counter = CnV), that is, at the end of the pulse width.

This type of PWM signal is called edge-aligned because the leading edges of all PWM signals are aligned with the beginning of the period, which is the same for all channels within an TPM.

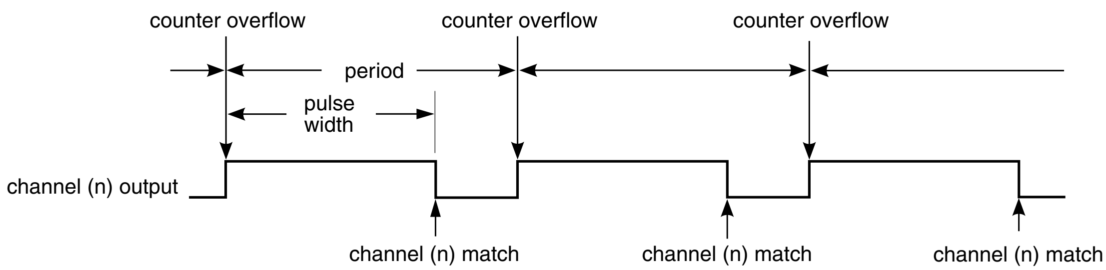


Figure 27-85. EPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1), however the channel (n) output is not controlled by TPM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the counter overflow (when the zero is loaded into the TPM counter), and it is forced low at the channel (n) match (TPM counter = CnV) (see the following figure).

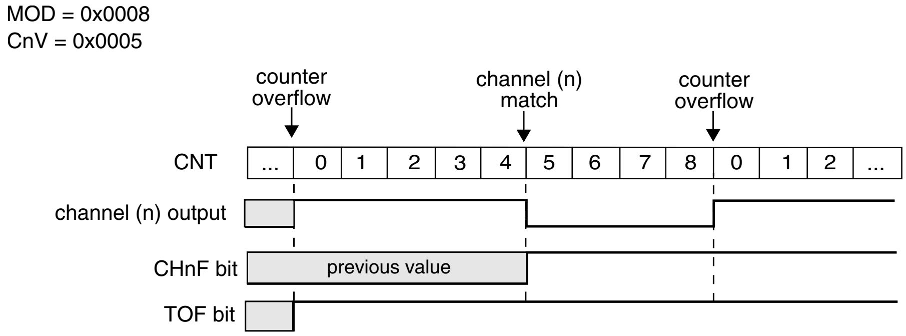


Figure 27-86. EPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the counter overflow (when zero is loaded into the TPM counter), and it is forced high at the channel (n) match (TPM counter = CnV) (see the following figure).

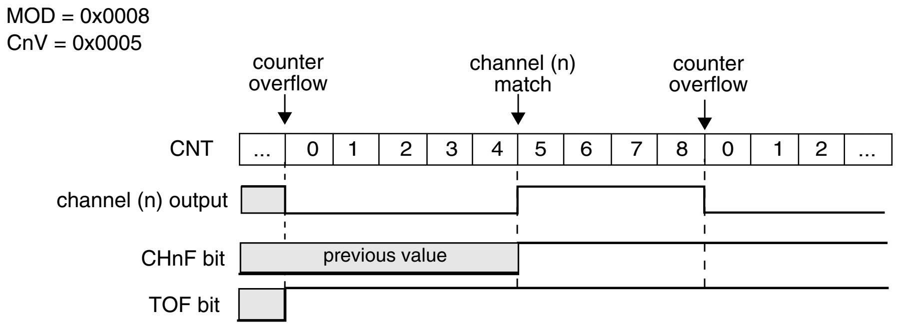


Figure 27-87. EPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000), then the channel (n) output is a 0% duty cycle EPWM signal. If (CnV > MOD), then the channel (n) output is a 100% duty cycle EPWM signal and CHnF bit is not set since there is never a channel (n) match. Therefore, MOD must be less than 0xFFFF in order to get a 100% duty cycle EPWM signal.

27.4.7 Center-Aligned PWM (CPWM) Mode

The center-aligned mode is selected when (CPWMS = 1) and (MSnB:MSnA = 1:0).

The CPWM pulse width (duty cycle) is determined by 2 × CnV and the period is determined by 2 × MOD (see the following figure). MOD must be kept in the range of 0x0001 to 0x7FFF because values outside this range can produce ambiguous results.

In the CPWM mode, the TPM counter counts up until it reaches MOD and then counts down until it reaches zero.

The CHnF bit is set and channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (TPM counter = CnV) when the TPM counting is down (at the begin of the pulse width) and when the TPM counting is up (at the end of the pulse width).

This type of PWM signal is called center-aligned because the pulse width centers for all channels are when the TPM counter is zero.

The other channel modes are not designed to be used with the up-down counter (CPWMS = 1). Therefore, all TPM channels should be used in CPWM mode when (CPWMS = 1).

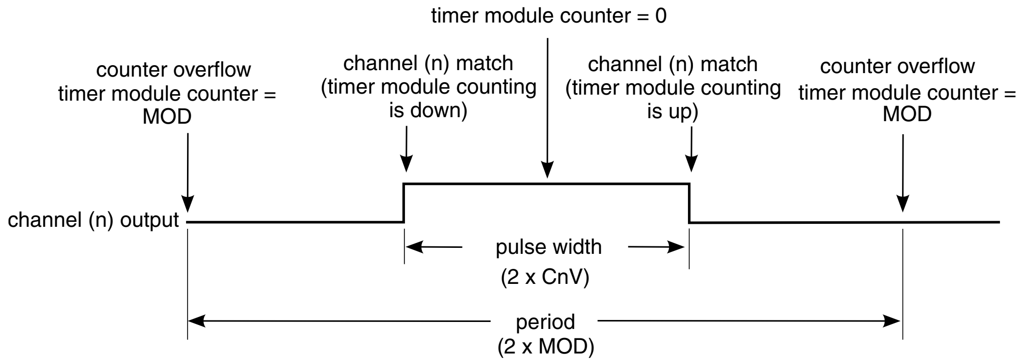


Figure 27-88. CPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the TPM counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1), however the channel (n) output is not controlled by TPM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the channel (n) match (TPM counter = CnV) when counting down, and it is forced low at the channel (n) match when counting up (see the following figure).

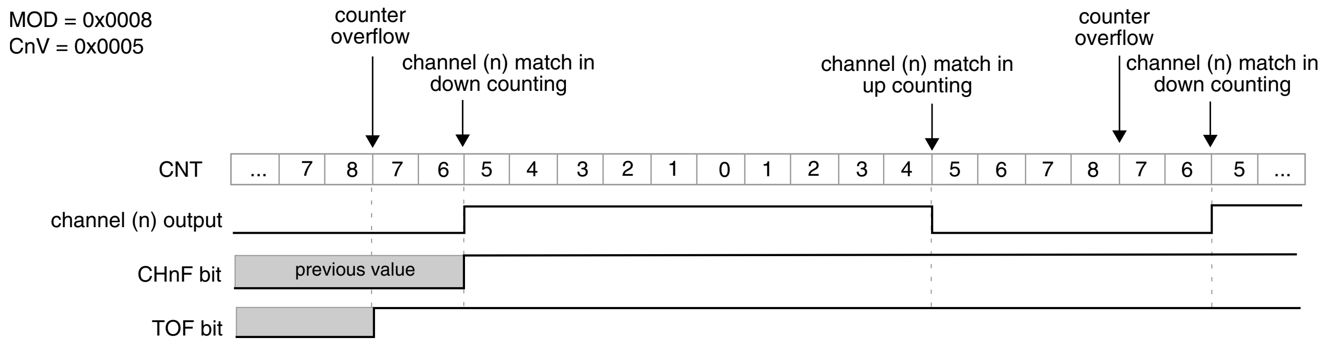


Figure 27-89. CPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the channel (n) match (TPM counter = CnV) when counting down, and it is forced high at the channel (n) match when counting up (see the following figure).

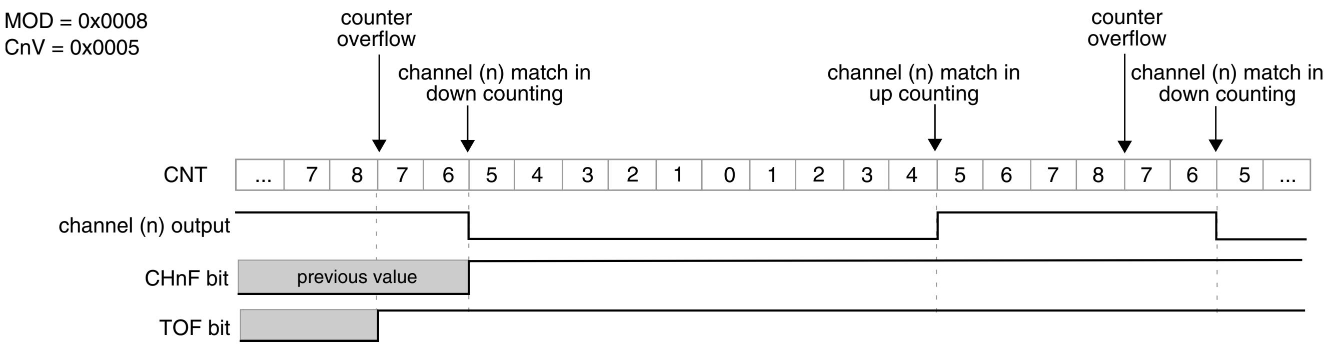


Figure 27-90. CPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000) then the channel (n) output is a 0% duty cycle CPWM signal.

If (CnV > MOD), then the channel (n) output is a 100% duty cycle CPWM signal, although the CHnF bit is set when the counter changes from incrementing to decrementing. Therefore, MOD must be less than 0xFFFF in order to get a 100% duty cycle CPWM signal.

27.4.8 Registers Updated from Write Buffers

27.4.8.1 MOD Register Update

If (CMOD[1:0] = 0:0) then MOD register is updated when MOD register is written.

If (CMOD[1:0] ≠ 0:0), then MOD register is updated according to the CPWMS bit, that is:

- If the selected mode is not CPWM then MOD register is updated after MOD register was written and the TPM counter changes from MOD to zero.
- If the selected mode is CPWM then MOD register is updated after MOD register was written and the TPM counter changes from MOD to (MOD – 1).

27.4.8.2 CnV Register Update

If (CMOD[1:0] = 0:0) then CnV register is updated when CnV register is written.

If (CMOD[1:0] ≠ 0:0), then CnV register is updated according to the selected mode, that is:

- If the selected mode is output compare then CnV register is updated on the next TPM counter increment (end of the prescaler counting) after CnV register was written.
- If the selected mode is EPWM then CnV register is updated after CnV register was written and the TPM counter changes from MOD to zero.
- If the selected mode is CPWM then CnV register is updated after CnV register was written and the TPM counter changes from MOD to (MOD – 1).

27.4.9 DMA

The channel generates a DMA transfer request according to DMA and CHnIE bits.

See the following table for more information.

Table 27-110. 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 generated if (CHnF = 1).	The channel interrupt is not generated.
1	1	The channel DMA transfer request is generated if (CHnF = 1).	The channel interrupt is generated if (CHnF = 1).

If DMA = 1, the CHnF bit can be cleared either by channel DMA transfer done or writing a one to CHnF bit (see the following table).

Table 27-111. Clear CHnF Bit

DMA	How CHnF Bit Can Be Cleared
0	CHnF bit is cleared by writing a 1 to CHnF bit.
1	CHnF bit is cleared either when the channel DMA transfer is done or by writing a 1 to CHnF bit.

27.4.10 Output triggers

The TPM generates output triggers for the counter and each channel that can be used to trigger events in other peripherals.

The counter trigger asserts whenever the TOF is set and remains asserted until the next increment.

Each TPM channel generates both a pre-trigger output and a trigger output. The pre-trigger output asserts whenever the CHnF is set, the trigger output asserts on the first counter increment after the pre-trigger asserts, and then both the trigger and pre-trigger negate on the first counter increment after the trigger asserts.

27.4.11 Reset Overview

The TPM is reset whenever any chip reset occurs.

When the TPM exits from reset:

- the TPM counter and the prescaler counter are zero and are stopped (CMOD[1:0] = 0:0);
- the timer overflow interrupt is zero;
- the channels interrupts are zero;
- the channels are in input capture mode;
- the channels outputs are zero;
- the channels pins are not controlled by TPM (ELS(n)B:ELS(n)A = 0:0).

27.4.12 TPM Interrupts

This section describes TPM interrupts.

27.4.12.1 Timer Overflow Interrupt

The timer overflow interrupt is generated when (TOIE = 1) and (TOF = 1).

27.4.12.2 Channel (n) Interrupt

The channel (n) interrupt is generated when (CHnIE = 1) and (CHnF = 1).

Chapter 28
Periodic Interrupt Timer (PIT)

28.1 Introduction

NOTE

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

The PIT module is an array of timers that can be used to raise interrupts and trigger DMA channels.

28.1.1 Block diagram

The following figure shows the block diagram of the PIT module.

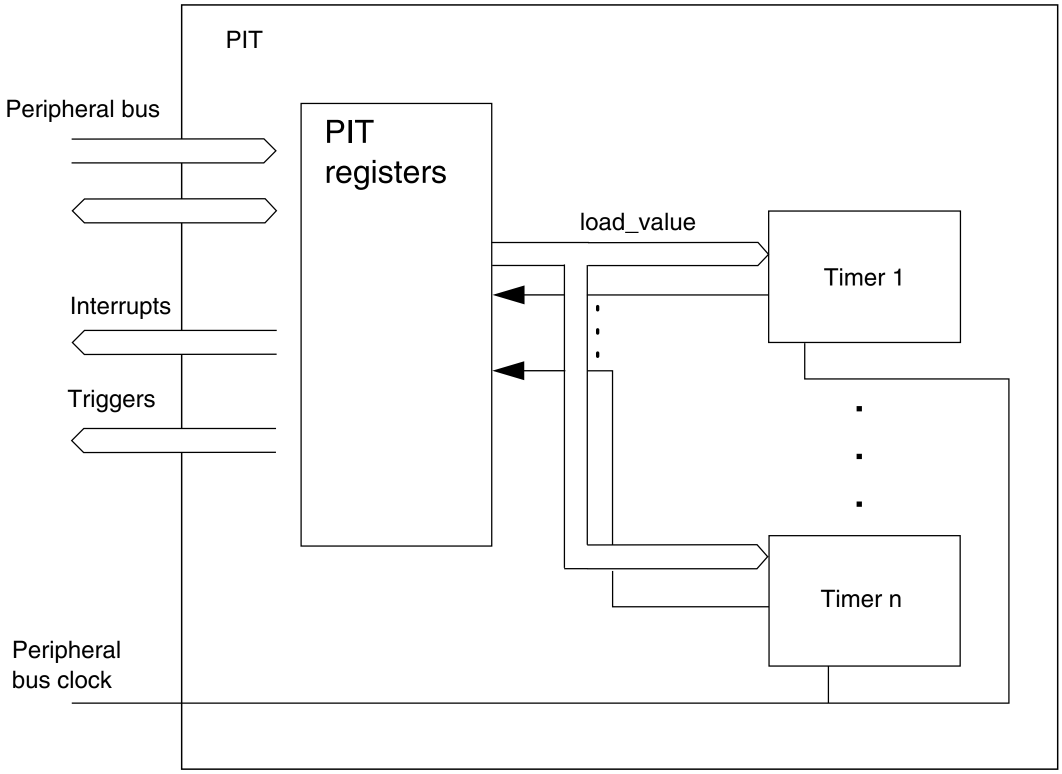


Figure 28-1. Block diagram of the PIT

NOTE

See the chip-specific PIT information for the number of PIT channels used in this MCU.

28.1.2 Features

The main features of this block are:

- Ability of timers to generate DMA trigger pulses
- Ability of timers to generate interrupts
- Maskable interrupts
- Independent timeout periods for each timer

28.2 Signal description

The PIT module has no external pins.

28.3 Memory map/register description

This section provides a detailed description of all registers accessible in the PIT module.

- Reserved registers will read as 0, writes will have no effect.
- See the chip-specific PIT information for the number of PIT channels used in this MCU.

PIT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_7000	PIT Module Control Register (PIT_MCR)	32	R/W	0000_0006h	28.3.1/531
4003_70E0	PIT Upper Lifetime Timer Register (PIT_LTMR64H)	32	R	0000_0000h	28.3.2/532
4003_70E4	PIT Lower Lifetime Timer Register (PIT_LTMR64L)	32	R	0000_0000h	28.3.3/533
4003_7100	Timer Load Value Register (PIT_LDVAL0)	32	R/W	0000_0000h	28.3.4/533
4003_7104	Current Timer Value Register (PIT_CVAL0)	32	R	0000_0000h	28.3.5/534
4003_7108	Timer Control Register (PIT_TCTRL0)	32	R/W	0000_0000h	28.3.6/534
4003_710C	Timer Flag Register (PIT_TFLG0)	32	R/W	0000_0000h	28.3.7/535
4003_7110	Timer Load Value Register (PIT_LDVAL1)	32	R/W	0000_0000h	28.3.4/533
4003_7114	Current Timer Value Register (PIT_CVAL1)	32	R	0000_0000h	28.3.5/534
4003_7118	Timer Control Register (PIT_TCTRL1)	32	R/W	0000_0000h	28.3.6/534
4003_711C	Timer Flag Register (PIT_TFLG1)	32	R/W	0000_0000h	28.3.7/535

28.3.1 PIT Module Control Register (PIT_MCR)

This register enables or disables the PIT timer clocks and controls the timers when the PIT enters the Debug mode.

Address: 4003_7000h base + 0h offset = 4003_7000h

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

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

PIT_MCR field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved.
1 MDIS	Module Disable - (PIT section) Disables the standard timers. This field must be enabled before any other setup is done. 0 Clock for standard PIT timers is enabled. 1 Clock for standard PIT timers is disabled.
0 FRZ	Freeze Allows the timers to be stopped when the device enters the Debug mode. 0 Timers continue to run in Debug mode. 1 Timers are stopped in Debug mode.

28.3.2 PIT Upper Lifetime Timer Register (PIT_LTMR64H)

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit lifetimer.

Address: 4003_7000h base + E0h offset = 4003_70E0h

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

PIT_LTMR64H field descriptions

Field	Description
31–0 LTH	Life Timer value Shows the timer value of timer 1. If this register is read at a time t1, LTMR64L shows the value of timer 0 at time t1.

28.3.3 PIT Lower Lifetime Timer Register (PIT_LTMR64L)

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit lifetimer.

To use LTMR64H and LTMR64L, timer 0 and timer 1 need to be chained. To obtain the correct value, first read LTMR64H and then LTMR64L. LTMR64H will have the value of CVAL1 at the time of the first access, LTMR64L will have the value of CVAL0 at the time of the first access, therefore the application does not need to worry about carry-over effects of the running counter.

Address: 4003_7000h base + E4h offset = 4003_70E4h

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

PIT_LTMR64L field descriptions

Field	Description
31–0 LTL	Life Timer value Shows the value of timer 0 at the time LTMR64H was last read. It will only update if LTMR64H is read.

28.3.4 Timer Load Value Register (PIT_LDVAL*n*)

These registers select the timeout period for the timer interrupts.

Address: 4003_7000h base + 100h offset + (16d × i), where i=0d to 1d

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

PIT_LDVALn field descriptions

Field	Description
31–0 TSV	Timer Start Value Sets the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer; instead the value will be loaded after the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again.

28.3.5 Current Timer Value Register (PIT_CVAL*n*)

These registers indicate the current timer position.

Address: 4003_7000h base + 104h offset + (16d × i), where i=0d to 1d

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

PIT_CVALn field descriptions

Field	Description
31–0 TVL	Current Timer Value Represents the current timer value, if the timer is enabled. **NOTE:** • If the timer is disabled, do not use this field as its value is unreliable. • The timer uses a downcounter. The timer values are frozen in Debug mode if MCR[FRZ] is set.

28.3.6 Timer Control Register (PIT_TCTRL*n*)

These registers contain the control bits for each timer.

Address: 4003_7000h base + 108h offset + (16d × i), where i=0d to 1d

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

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

PIT_TCTRL*n* field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CHN	Chain Mode When activated, Timer n-1 needs to expire before timer n can decrement by 1. Timer 0 cannot be chained. 0 Timer is not chained. 1 Timer is chained to previous timer. For example, for Channel 2, if this field is set, Timer 2 is chained to Timer 1.
1 TIE	Timer Interrupt Enable When an interrupt is pending, or, TFLGn[TIF] is set, enabling the interrupt will immediately cause an interrupt event. To avoid this, the associated TFLGn[TIF] must be cleared first. 0 Interrupt requests from Timer n are disabled. 1 Interrupt will be requested whenever TIF is set.
0 TEN	Timer Enable Enables or disables the timer. 0 Timer n is disabled. 1 Timer n is enabled.

28.3.7 Timer Flag Register (PIT_TFLG*n*)

These registers hold the PIT interrupt flags.

Address: 4003_7000h base + 10Ch offset + (16d × i), where i=0d to 1d

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															TIF
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_TFLG*n* field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

PIT_TFLG*n* field descriptions (continued)

Field	Description
0 TIF	Timer Interrupt Flag Sets to 1 at the end of the timer period. Writing 1 to this flag clears it. Writing 0 has no effect. If enabled, or, when TCTRLn[TIE] = 1, TIF causes an interrupt request. 0 Timeout has not yet occurred. 1 Timeout has occurred.

28.4 Functional description

This section provides the functional description of the module.

28.4.1 General operation

This section gives detailed information on the internal operation of the module. Each timer can be used to generate trigger pulses and interrupts. Each interrupt is available on a separate interrupt line.

28.4.1.1 Timers

The timers generate triggers at periodic intervals, when enabled. The timers load the start values as specified in their LDVAL registers, count down to 0 and then load the respective start value again. Each time a timer reaches 0, it will generate a trigger pulse and set the interrupt flag.

All interrupts can be enabled or masked by setting TCTRLn[TIE]. A new interrupt can be generated only after the previous one is cleared.

If desired, the current counter value of the timer can be read via the CVAL registers.

The counter period can be restarted, by first disabling, and then enabling the timer with TCTRLn[TEN]. See the following figure.

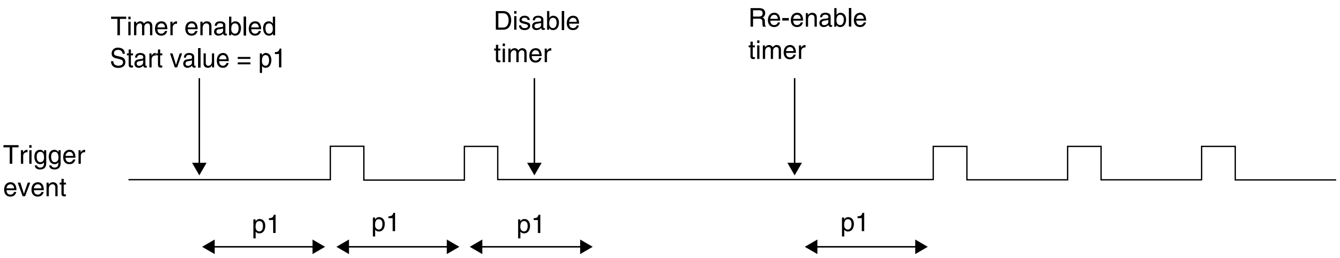


Figure 28-17. Stopping and starting a timer

The counter period of a running timer can be modified, by first disabling the timer, setting a new load value, and then enabling the timer again. See the following figure.

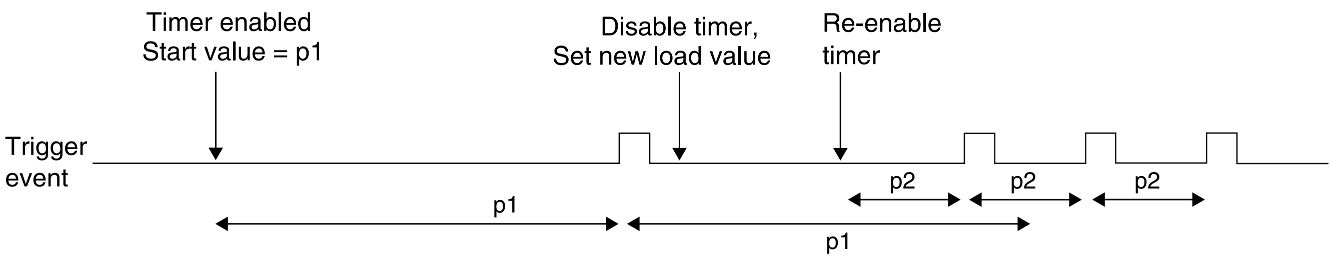


Figure 28-18. Modifying running timer period

It is also possible to change the counter period without restarting the timer by writing LDVAL with the new load value. This value will then be loaded after the next trigger event. See the following figure.

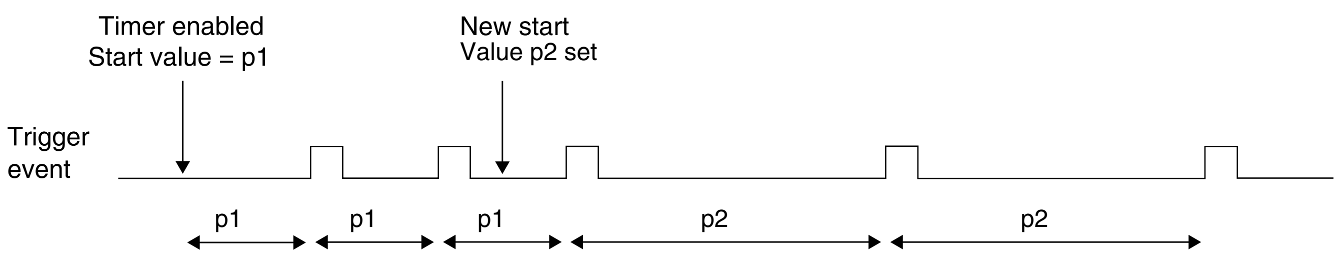


Figure 28-19. Dynamically setting a new load value

28.4.1.2 Debug mode

In Debug mode, the timers will be frozen based on MCR[FRZ]. This is intended to aid software development, allowing the developer to halt the processor, investigate the current state of the system, for example, the timer values, and then continue the operation.

28.4.2 Interrupts

All the timers support interrupt generation. See the MCU specification for related vector addresses and priorities.

Timer interrupts can be enabled by setting TCTRLn[TIE]. TFLGn[TIF] are set to 1 when a timeout occurs on the associated timer, and are cleared to 0 by writing a 1 to the corresponding TFLGn[TIF].

28.4.3 Chained timers

When a timer has chain mode enabled, it will only count after the previous timer has expired. So if timer n-1 has counted down to 0, counter n will decrement the value by one. This allows to chain some of the timers together to form a longer timer. The first timer (timer 0) cannot be chained to any other timer.

28.5 Initialization and application information

In the example configuration:

- The PIT clock has a frequency of 50 MHz.
- Timer 1 creates an interrupt every 5.12 ms.
- Timer 3 creates a trigger event every 30 ms.

The PIT module must be activated by writing a 0 to MCR[MDIS].

The 50 MHz clock frequency equates to a clock period of 20 ns. Timer 1 needs to trigger every 5.12 ms/20 ns = 256,000 cycles and Timer 3 every 30 ms/20 ns = 1,500,000 cycles. The value for the LDVAL register trigger is calculated as:

LDVAL trigger = (period / clock period) -1

This means LDVAL1 and LDVAL3 must be written with 0x0003E7FF and 0x0016E35F respectively.

The interrupt for Timer 1 is enabled by setting TCTRL1[TIE]. The timer is started by writing 1 to TCTRL1[TEN].

Timer 3 shall be used only for triggering. Therefore, Timer 3 is started by writing a 1 to TCTRL3[TEN]. TCTRL3[TIE] stays at 0.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 1
PIT_LDVAL1 = 0x0003E7FF; // setup timer 1 for 256000 cycles
PIT_TCTRL1 = TIE; // enable Timer 1 interrupts
PIT_TCTRL1 |= TEN; // start Timer 1


// Timer 3
PIT_LDVAL3 = 0x0016E35F; // setup timer 3 for 1500000 cycles
PIT_TCTRL3 |= TEN; // start Timer 3
```

28.6 Example configuration for chained timers

In the example configuration:

- The PIT clock has a frequency of 100 MHz.
- Timers 1 and 2 are available.
- An interrupt shall be raised every 1 hour.

The PIT module needs to be activated by writing a 0 to MCR[MDIS].

The 100 MHz clock frequency equates to a clock period of 10 ns, so the PIT needs to count for 6000 million cycles, which is more than a single timer can do. So, Timer 1 is set up to trigger every 6 s (600 million cycles). Timer 2 is chained to Timer 1 and programmed to trigger 10 times.

The value for the LDVAL register trigger is calculated as number of cycles-1, so LDVAL1 receives the value 0x23C345FF and LDVAL2 receives the value 0x00000009.

The interrupt for Timer 2 is enabled by setting TCTRL2[TIE], the Chain mode is activated by setting TCTRL2[CHN], and the timer is started by writing a 1 to TCTRL2[TEN]. TCTRL1[TEN] needs to be set, and TCTRL1[CHN] and TCTRL1[TIE] are cleared.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 2
PIT_LDVAL2 = 0x00000009; // setup Timer 2 for 10 counts
PIT_TCTRL2 = TIE; // enable Timer 2 interrupt
PIT_TCTRL2 |= CHN; // chain Timer 2 to Timer 1
PIT_TCTRL2 |= TEN; // start Timer 2
```

```
// Timer 1
PIT_LDVAL1 = 0x23C345FF; // setup Timer 1 for 600 000 000 cycles
PIT_TCTRL1 = TEN; // start Timer 1
```

28.7 Example configuration for the lifetime timer

To configure the lifetimer timer, channels 0 and 1 need to be chained together.

First the PIT module needs to be activated by writing a 0 to the MDIS bit in the CTRL register, then the LDVAL registers need to be set to the maximum value.

The timer is a downcounter.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 1
PIT_LDVAL1 = 0xFFFFFFFF; // setup timer 1 for maximum counting period
PIT_TCTRL1 = 0x0; // disable timer 1 interrupts
PIT_TCTRL1 |= CHN; // chain timer 1 to timer 0
PIT_TCTRL1 |= TEN; // start timer 1


// Timer 0
PIT_LDVAL0 = 0xFFFFFFFF; // setup timer 0 for maximum counting period
PIT_TCTRL0 = TEN; // start timer 0
```

To access the lifetime, read first LTMR64H and then LTMR64L.

```
current_uptime = PIT_LTMR64H<<32;
current_uptime = current_uptime + PIT_LTMR64L;
```

Chapter 29
Low-Power Timer (LPTMR)

29.1 Introduction

NOTE

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

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.

29.1.1 Features

The features of the LPTMR module include:

- 16-bit time counter or pulse counter with compare
    - Optional interrupt can generate asynchronous wakeup from any low-power mode
    - Hardware trigger output
    - Counter supports free-running mode or reset on compare
- Configurable clock source for prescaler/glitch filter
- Configurable input source for pulse counter
    - Rising-edge or falling-edge

29.1.2 Modes of operation

The following table describes the operation of the LPTMR module in various modes.

Table 29-1. Modes of operation

Modes	Description
Run	The LPTMR operates normally.
Wait	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Stop	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Low-Leakage	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Debug	The LPTMR operates normally in Pulse Counter mode, but counter does not increment in Time Counter mode.

29.2 LPTMR signal descriptions

Table 29-2. LPTMR signal descriptions

Signal	I/O	Description
LPTMR_ALT*n*	I	Pulse Counter Input pin

29.2.1 Detailed signal descriptions

Table 29-3. LPTMR interface—detailed signal descriptions

<table>
<tr><th>Signal</th><th>I/O</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">LPTMR_ALTn</td><td rowspan="3">I</td><td colspan="2">Pulse Counter Input
The LPTMR can select one of the input pins to be used in Pulse Counter mode.</td></tr>
<tr><td>State meaning</td><td>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.</td></tr>
<tr><td>Timing</td><td>Assertion or deassertion may occur at any time; input may assert asynchronously to the bus clock.</td></tr>
</table>

29.3 Memory map and register definition

LPTMR memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_0000	Low Power Timer Control Status Register (LPTMR0_CSR)	32	R/W	0000_0000h	29.3.1/543
4004_0004	Low Power Timer Prescale Register (LPTMR0_PSR)	32	R/W	0000_0000h	29.3.2/544
4004_0008	Low Power Timer Compare Register (LPTMR0_CMR)	32	R/W	0000_0000h	29.3.3/546
4004_000C	Low Power Timer Counter Register (LPTMR0_CNR)	32	R	0000_0000h	29.3.4/546

29.3.1 Low Power Timer Control Status Register (LPTMRx_CSR)

Address: 4004_0000h base + 0h offset = 4004_0000h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TCF	TIE	TPS		TPP	TFC	TMS	TEN
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CSR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TCF	Timer Compare Flag TCF is set when the LPTMR is enabled and the CNR equals the CMR and increments. TCF is cleared when the LPTMR is disabled or a logic 1 is written to it. 0 The value of CNR is not equal to CMR and increments. 1 The value of CNR is equal to CMR and increments.
6 TIE	Timer Interrupt Enable When TIE is set, the LPTMR Interrupt is generated whenever TCF is also set. 0 Timer interrupt disabled. 1 Timer interrupt enabled.
5–4 TPS	Timer Pin Select Configures the input source to be used in Pulse Counter mode. TPS must be altered only when the LPTMR is disabled. The input connections vary by device. See the chip configuration details for information on the connections to these inputs. 00 Pulse counter input 0 is selected.

Table continues on the next page...

LPTMRx_CSR field descriptions (continued)

Field	Description
	01 Pulse counter input 1 is selected. 10 Pulse counter input 2 is selected. 11 Pulse counter input 3 is selected.
3 TPP	Timer Pin Polarity Configures the polarity of the input source in Pulse Counter mode. TPP must be changed only when the LPTMR is disabled. 0 Pulse Counter input source is active-high, and the CNR will increment on the rising-edge. 1 Pulse Counter input source is active-low, and the CNR will increment on the falling-edge.
2 TFC	Timer Free-Running Counter When clear, TFC configures the CNR to reset whenever TCF is set. When set, TFC configures the CNR to reset on overflow. TFC must be altered only when the LPTMR is disabled. 0 CNR is reset whenever TCF is set. 1 CNR is reset on overflow.
1 TMS	Timer Mode Select Configures the mode of the LPTMR. TMS must be altered only when the LPTMR is disabled. 0 Time Counter mode. 1 Pulse Counter mode.
0 TEN	Timer Enable When TEN is clear, it resets the LPTMR internal logic, including the CNR and TCF. When TEN is set, the LPTMR is enabled. While writing 1 to this field, CSR[5:1] must not be altered. 0 LPTMR is disabled and internal logic is reset. 1 LPTMR is enabled.

29.3.2 Low Power Timer Prescale Register (LPTMRx_PSR)

Address: 4004_0000h base + 4h offset = 4004_0004h

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

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

LPTMRx_PSR field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–3 PRESCALE	Prescale Value Configures the size of the Prescaler in Time Counter mode or width of the glitch filter in Pulse Counter mode. PRESCALE must be altered only when the LPTMR is disabled. 0000 Prescaler divides the prescaler clock by 2; glitch filter does not support this configuration. 0001 Prescaler divides the prescaler clock by 4; glitch filter recognizes change on input pin after 2 rising clock edges. 0010 Prescaler divides the prescaler clock by 8; glitch filter recognizes change on input pin after 4 rising clock edges. 0011 Prescaler divides the prescaler clock by 16; glitch filter recognizes change on input pin after 8 rising clock edges. 0100 Prescaler divides the prescaler clock by 32; glitch filter recognizes change on input pin after 16 rising clock edges. 0101 Prescaler divides the prescaler clock by 64; glitch filter recognizes change on input pin after 32 rising clock edges. 0110 Prescaler divides the prescaler clock by 128; glitch filter recognizes change on input pin after 64 rising clock edges. 0111 Prescaler divides the prescaler clock by 256; glitch filter recognizes change on input pin after 128 rising clock edges. 1000 Prescaler divides the prescaler clock by 512; glitch filter recognizes change on input pin after 256 rising clock edges. 1001 Prescaler divides the prescaler clock by 1024; glitch filter recognizes change on input pin after 512 rising clock edges. 1010 Prescaler divides the prescaler clock by 2048; glitch filter recognizes change on input pin after 1024 rising clock edges. 1011 Prescaler divides the prescaler clock by 4096; glitch filter recognizes change on input pin after 2048 rising clock edges. 1100 Prescaler divides the prescaler clock by 8192; glitch filter recognizes change on input pin after 4096 rising clock edges. 1101 Prescaler divides the prescaler clock by 16,384; glitch filter recognizes change on input pin after 8192 rising clock edges. 1110 Prescaler divides the prescaler clock by 32,768; glitch filter recognizes change on input pin after 16,384 rising clock edges. 1111 Prescaler divides the prescaler clock by 65,536; glitch filter recognizes change on input pin after 32,768 rising clock edges.
2 PBYP	Prescaler Bypass When PBYP is set, the selected prescaler clock in Time Counter mode or selected input source in Pulse Counter mode directly clocks the CNR. When PBYP is clear, the CNR is clocked by the output of the prescaler/glitch filter. PBYP must be altered only when the LPTMR is disabled. 0 Prescaler/glitch filter is enabled. 1 Prescaler/glitch filter is bypassed.
1–0 PCS	Prescaler Clock Select Selects the clock to be used by the LPTMR prescaler/glitch filter. PCS must be altered only when the LPTMR is disabled. The clock connections vary by device. **NOTE:** See the chip configuration details for information on the connections to these inputs.

Table continues on the next page...

LPTMRx_PSR field descriptions (continued)

Field	Description
	00 Prescaler/glitch filter clock 0 selected. 01 Prescaler/glitch filter clock 1 selected. 10 Prescaler/glitch filter clock 2 selected. 11 Prescaler/glitch filter clock 3 selected.

29.3.3 Low Power Timer Compare Register (LPTMRx_CMR)

Address: 4004_0000h base + 8h offset = 4004_0008h

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

LPTMRx_CMR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 COMPARE	Compare Value When the LPTMR is enabled and the CNR equals the value in the CMR and increments, TCF is set and the hardware trigger asserts until the next time the CNR increments. If the CMR is 0, the hardware trigger will remain asserted until the LPTMR is disabled. If the LPTMR is enabled, the CMR must be altered only when TCF is set.

29.3.4 Low Power Timer Counter Register (LPTMRx_CNR)

Address: 4004_0000h base + Ch offset = 4004_000Ch

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

LPTMRx_CNR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 COUNTER	Counter Value

29.4 Functional description

29.4.1 LPTMR power and reset

The LPTMR remains powered in all power modes, including low-leakage modes. If the LPTMR is not required to remain operating during a low-power mode, then it must be disabled before entering the mode.

The LPTMR is reset only on global Power On Reset (POR) or Low Voltage Detect (LVD). When configuring the LPTMR registers, the CSR must be initially written with the timer disabled, before configuring the PSR and CMR. Then, CSR[TIE] must be set as the last step in the initialization. This ensures the LPTMR is configured correctly and the LPTMR counter is reset to zero following a warm reset.

29.4.2 LPTMR clocking

The LPTMR prescaler/glitch filter can be clocked by one of the four clocks. The clock source must be enabled before the LPTMR is enabled.

NOTE

The clock source selected 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.

29.4.3 LPTMR prescaler/glitch filter

The LPTMR prescaler and glitch filter share the same logic which operates as a prescaler in Time Counter mode and as a glitch filter in Pulse Counter mode.

NOTE

The prescaler/glitch filter configuration must not be altered when the LPTMR is enabled.

29.4.3.1 Prescaler enabled

In Time Counter mode, when the prescaler is enabled, the output of the prescaler directly clocks the CNR. When the LPTMR is enabled, the CNR will increment every 2^2 to 2^{16} prescaler clock cycles. After the LPTMR is enabled, the first increment of the CNR will take an additional one or two prescaler clock cycles due to synchronization logic.

29.4.3.2 Prescaler bypassed

In Time Counter mode, when the prescaler is bypassed, the selected prescaler clock increments the CNR on every clock cycle. When the LPTMR is enabled, the first increment will take an additional one or two prescaler clock cycles due to synchronization logic.

29.4.3.3 Glitch filter

In Pulse Counter mode, when the glitch filter is enabled, the output of the glitch filter directly clocks the CNR. When the LPTMR is first enabled, the output of the glitch filter is asserted, that is, logic 1 for active-high and logic 0 for active-low. The following table shows the change in glitch filter output with the selected input source.

If	Then
The selected input source remains deasserted for at least 2^1 to 2^{15} consecutive prescaler clock rising edges	The glitch filter output will also deassert.
The selected input source remains asserted for at least 2^1 to 2^{15} consecutive prescaler clock rising-edges	The glitch filter output will also assert.

NOTE

The input is only sampled on the rising clock edge.

The CNR will increment each time the glitch filter output asserts. In Pulse Counter mode, the maximum rate at which the CNR can increment is once every 2^2 to 2^{16} prescaler clock edges. When first enabled, the glitch filter will wait an additional one or two prescaler clock edges due to synchronization logic.

29.4.3.4 Glitch filter bypassed

In Pulse Counter mode, when the glitch filter is bypassed, the selected input source increments the CNR every time it asserts. Before the LPTMR is first enabled, the selected input source is forced to be asserted. This prevents the CNR from incrementing if the selected input source is already asserted when the LPTMR is first enabled.

29.4.4 LPTMR compare

When the CNR equals the value of the CMR and increments, the following events occur:

- CSR[TCF] is set.
- LPTMR interrupt is generated if CSR[TIE] is also set.
- LPTMR hardware trigger is generated.
- CNR is reset if CSR[TFC] is clear.

When the LPTMR is enabled, the CMR can be altered only when CSR[TCF] is set. When updating the CMR, the CMR must be written and CSR[TCF] must be cleared before the LPTMR counter has incremented past the new LPTMR compare value.

29.4.5 LPTMR counter

The CNR increments by one on every:

- Prescaler clock in Time Counter mode with prescaler bypassed
- Prescaler output in Time Counter mode with prescaler enabled
- Input source assertion in Pulse Counter mode with glitch filter bypassed
- Glitch filter output in Pulse Counter mode with glitch filter enabled

The CNR is reset when the LPTMR is disabled or if the counter register overflows. If CSR[TFC] is cleared, then the CNR is also reset whenever CSR[TCF] is set.

The CNR continues incrementing when the core is halted in Debug mode when configured for Pulse Counter mode, the CNR will stop incrementing when the core is halted in Debug mode when configured for Time Counter mode.

The CNR cannot be initialized, but can be read at any time. On each read of the CNR, software must first write to the CNR with any value. This will synchronize and register the current value of the CNR into a temporary register. The contents of the temporary register are returned on each read of the CNR.

When reading the CNR, the bus clock must be at least two times faster than the rate at which the LPTMR counter is incrementing, otherwise incorrect data may be returned.

29.4.6 LPTMR hardware trigger

The LPTMR hardware trigger asserts at the same time the CSR[TCF] is set and can be used to trigger hardware events in other peripherals without software intervention. The hardware trigger is always enabled.

When	Then
The CMR is set to 0 with CSR[TFC] clear	The LPTMR hardware trigger will assert on the first compare and does not deassert.
The CMR is set to a nonzero value, or, if CSR[TFC] is set	The LPTMR hardware trigger will assert on each compare and deassert on the following increment of the CNR.

29.4.7 LPTMR interrupt

The LPTMR interrupt is generated whenever CSR[TIE] and CSR[TCF] are set. CSR[TCF] is cleared by disabling the LPTMR or by writing a logic 1 to it.

CSR[TIE] can be altered and CSR[TCF] can be cleared while the LPTMR is enabled.

The LPTMR interrupt is generated asynchronously to the system clock and can be used to generate a wakeup from any low-power mode, including the low-leakage modes, provided the LPTMR is enabled as a wakeup source.

Chapter 30
Real Time Clock (RTC)

30.1 Introduction

NOTE

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

30.1.1 Features

The RTC module features include:

- 32-bit seconds counter with roll-over protection and 32-bit alarm
- 16-bit prescaler with compensation that can correct errors between 0.12 ppm and 3906 ppm
- Register write protection
    - Lock register requires POR or software reset to enable write access
- 1 Hz square wave output

30.1.2 Modes of operation

The RTC remains functional in all low power modes and can generate an interrupt to exit any low power mode.

30.1.3 RTC signal descriptions

Table 30-1. RTC signal descriptions

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

30.1.3.1 RTC clock output

The clock to the seconds counter is available on the RTC_CLKOUT signal. It is a 1 Hz square wave output.

30.2 Register definition

All registers must be accessed using 32-bit writes and all register accesses incur three wait states.

Write accesses to any register by non-supervisor mode software, when the supervisor access bit in the control register is clear, will terminate with a bus error.

Read accesses by non-supervisor mode software complete as normal.

Writing to a register protected by the lock register does not generate a bus error, but the write will not complete.

RTC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_D000	RTC Time Seconds Register (RTC_TSR)	32	R/W	0000_0000h	30.2.1/553
4003_D004	RTC Time Prescaler Register (RTC_TPR)	32	R/W	0000_0000h	30.2.2/553
4003_D008	RTC Time Alarm Register (RTC_TAR)	32	R/W	0000_0000h	30.2.3/554
4003_D00C	RTC Time Compensation Register (RTC_TCR)	32	R/W	0000_0000h	30.2.4/554
4003_D010	RTC Control Register (RTC_CR)	32	R/W	0000_0000h	30.2.5/555
4003_D014	RTC Status Register (RTC_SR)	32	R/W	0000_0001h	30.2.6/557
4003_D018	RTC Lock Register (RTC_LR)	32	R/W	0000_00FFh	30.2.7/558
4003_D01C	RTC Interrupt Enable Register (RTC_IER)	32	R/W	0000_0007h	30.2.8/559

30.2.1 RTC Time Seconds Register (RTC_TSR)

Address: 4003_D000h base + 0h offset = 4003_D000h

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

RTC_TSR field descriptions

Field	Description
31–0 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).

30.2.2 RTC Time Prescaler Register (RTC_TPR)

Address: 4003_D000h base + 4h offset = 4003_D004h

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

RTC_TPR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 TPR	Time Prescaler Register When the time counter is enabled, the TPR is read only and increments every 32.768 kHz clock cycle. The time counter will read as zero when SR[TOF] or SR[TIF] are set. When the time counter is disabled, the TPR can be read or written. The TSR[TSR] increments when bit 14 of the TPR transitions from a logic one to a logic zero.

30.2.3 RTC Time Alarm Register (RTC_TAR)

Address: 4003_D000h base + 8h offset = 4003_D008h

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

RTC_TAR field descriptions

Field	Description
31–0 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].

30.2.4 RTC Time Compensation Register (RTC_TCR)

Address: 4003_D000h base + Ch offset = 4003_D00Ch

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

RTC_TCR field descriptions

Field	Description
31–24 CIC	Compensation Interval Counter Current value of the compensation interval counter. If the compensation interval counter equals zero then it is loaded with the contents of the CIR. If the CIC does not equal zero then it is decremented once a second.
23–16 TCV	Time Compensation Value Current value used by the compensation logic for the present second interval. Updated once a second if the CIC equals 0 with the contents of the TCR field. If the CIC does not equal zero then it is loaded with zero (compensation is not enabled for that second increment).
15–8 CIR	Compensation Interval Register Configures the compensation interval in seconds from 1 to 256 to control how frequently the TCR should adjust the number of 32.768 kHz cycles in each second. The value written should be one less than the number of seconds. For example, write zero to configure for a compensation interval of one second. This register is double buffered and writes do not take affect until the end of the current compensation interval.
7–0 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.

Table continues on the next page...

RTC_TCR field descriptions (continued)

Field	Description
	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.

30.2.5 RTC Control Register (RTC_CR)

Address: 4003_D000h base + 10h offset = 4003_D010h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	Reserved	SC2P	SC4P	SC8P	SC16P	CLKO	OSCE	0			WPS	UM	SUP	WPE	SWR
W		0														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_CR field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

RTC_CR field descriptions (continued)

Field	Description
14 Reserved	This field is reserved. It must always be written to 0.
13 SC2P	Oscillator 2pF Load Configure 0 Disable the load. 1 Enable the additional load.
12 SC4P	Oscillator 4pF Load Configure 0 Disable the load. 1 Enable the additional load.
11 SC8P	Oscillator 8pF Load Configure 0 Disable the load. 1 Enable the additional load.
10 SC16P	Oscillator 16pF Load Configure 0 Disable the load. 1 Enable the additional load.
9 CLKO	Clock Output 0 The 32 kHz clock is output to other peripherals. 1 The 32 kHz clock is not output to other peripherals.
8 OSCE	Oscillator Enable 0 32.768 kHz oscillator is disabled. 1 32.768 kHz oscillator is enabled. After setting this bit, wait the oscillator startup time before enabling the time counter to allow the 32.768 kHz clock time to stabilize.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 WPS	Wakeup Pin Select The wakeup pin is optional and not available on all devices. 0 Wakeup pin asserts (active low, open drain) if the RTC interrupt asserts or the wakeup pin is turned on. 1 Wakeup pin instead outputs the RTC 32kHz clock, provided the wakeup pin is turned on and the 32kHz clock is output to other peripherals.
3 UM	Update Mode Allows SR[TCE] to be written even when the Status Register is locked. When set, the SR[TCE] can always be written if the SR[TIF] or SR[TOF] are set or if the SR[TCE] is clear. 0 Registers cannot be written when locked. 1 Registers can be written when locked under limited conditions.
2 SUP	Supervisor Access 0 Non-supervisor mode write accesses are not supported and generate a bus error. 1 Non-supervisor mode write accesses are supported.
1 WPE	Wakeup Pin Enable

Table continues on the next page...

RTC_CR field descriptions (continued)

Field	Description
	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.

30.2.6 RTC Status Register (RTC_SR)

Address: 4003_D000h base + 14h offset = 4003_D014h

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

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

RTC_SR field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 TCE	Time Counter Enable When time counter is disabled the TSR register and TPR register are writeable, but do not increment. When time counter is enabled the TSR register and TPR register are not writeable, but increment. 0 Time counter is disabled. 1 Time counter is enabled.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 TAF	Time Alarm Flag Time alarm flag is set when the TAR[TAR] equals the TSR[TSR] and the TSR[TSR] increments. This bit is cleared by writing the TAR register. 0 Time alarm has not occurred. 1 Time alarm has occurred.

Table continues on the next page...

RTC_SR field descriptions (continued)

Field	Description
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.

30.2.7 RTC Lock Register (RTC_LR)

Address: 4003_D000h base + 18h offset = 4003_D018h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								1	LRL	SRL	CRL	TCL	1		
W										LRL	SRL	CRL	TCL			
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.

Table continues on the next page...

RTC_LR field descriptions (continued)

Field	Description
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.
2–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

30.2.8 RTC Interrupt Enable Register (RTC_IER)

Address: 4003_D000h base + 1Ch offset = 4003_D01Ch

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								WPON	Reserved		TSIE	Reserved	TAIE	TOIE	TIIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

RTC_IER field descriptions

Field	Description
31–8 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.

Table continues on the next page...

RTC_IER field descriptions (continued)

Field	Description
4 TSIE	Time Seconds Interrupt Enable The seconds interrupt is an edge-sensitive interrupt with a dedicated interrupt vector. It is generated once a second and requires no software overhead (there is no corresponding status flag to clear). 0 Seconds interrupt is disabled. 1 Seconds interrupt is enabled.
3 Reserved	This field is reserved.
2 TAIE	Time Alarm Interrupt Enable 0 Time alarm flag does not generate an interrupt. 1 Time alarm flag does generate an interrupt.
1 TOIE	Time Overflow Interrupt Enable 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.

30.3 Functional description

30.3.1 Power, clocking, and reset

The RTC is an always powered block that remains active in all low power modes.

The time counter within the RTC is clocked by a 32.768 kHz clock sourced from an external crystal using the oscillator.

The power-on-reset signal initializes all RTC registers to their default state. A software reset bit can also initialize all RTC registers.

30.3.1.1 Oscillator control

The 32.768 kHz crystal oscillator is disabled at POR and must be enabled by software. After enabling the cystal oscillator, wait the oscillator startup time before setting SR[TCE] or using the oscillator clock external to the RTC.

The crystal oscillator includes tunable capacitors that can be configured by software. Do not change the capacitance unless the oscillator is disabled.

30.3.1.2 Software reset

Writing 1 to CR[SWR] forces the equivalent of a POR to the rest of the RTC module. CR[SWR] is not affected by the software reset and must be cleared by software.

30.3.1.3 Supervisor access

When the supervisor access control bit is clear, only supervisor mode software can write to the RTC registers, non-supervisor mode software will generate a bus error. Both supervisor and non-supervisor mode software can always read the RTC registers.

30.3.2 Time counter

The time counter consists of a 32-bit seconds counter that increments once every second and a 16-bit prescaler register that increments once every 32.768 kHz clock cycle.

Reading the time counter (either seconds or prescaler) while is is incrementing may return invalid data due to synchronization of the read data bus. If it is necessary for software to read the prescaler or seconds counter when they could be incrementing, it is recommended that two read accesses are performed and that software verifies that the same data was returned for both reads.

The time seconds register and time prescaler register can be written only when SR[TCE] is clear. Always write to the prescaler register before writing to the seconds register, because the seconds register increments on the falling edge of bit 14 of the prescaler register.

The time prescaler register increments provided SR[TCE] is set, SR[TIF] is clear, SR[TOF] is clear, and the 32.768 kHz clock source is present. After enabling the oscillator, wait the oscillator startup time before setting SR[TCE] to allow time for the oscillator clock output to stabilize.

If the time seconds register overflows then the SR[TOF] will set and the time prescaler register will stop incrementing. Clear SR[TOF] by initializing the time seconds register. The time seconds register and time prescaler register read as zero whenever SR[TOF] is set.

SR[TIF] is set on POR and software reset and is cleared by initializing the time seconds register. The time seconds register and time prescaler register read as zero whenever SR[TIF] is set.

30.3.3 Compensation

The compensation logic provides an accurate and wide compensation range and can correct errors as high as 3906 ppm and as low as 0.12 ppm. The compensation factor must be calculated externally to the RTC and supplied by software to the compensation register. The RTC itself does not calculate the amount of compensation that is required, although the 1 Hz clock is output to an external pin in support of external calibration logic.

Crystal compensation can be supported by using firmware and crystal characteristics to determine the compensation amount. Temperature compensation can be supported by firmware that periodically measures the external temperature via ADC and updates the compensation register based on a look-up table that specifies the change in crystal frequency over temperature.

The compensation logic alters the number of 32.768 kHz clock cycles it takes for the prescaler register to overflow and increment the time seconds counter. The time compensation value is used to adjust the number of clock cycles between -127 and +128. Cycles are added or subtracted from the prescaler register when the prescaler register equals 0x3FFF and then increments. The compensation interval is used to adjust the frequency at which the time compensation value is used, that is, from once a second to once every 256 seconds.

Updates to the time compensation register will not take effect until the next time the time seconds register increments and provided the previous compensation interval has expired. When the compensation interval is set to other than once a second then the compensation is applied in the first second interval and the remaining second intervals receive no compensation.

Compensation is disabled by configuring the time compensation register to zero.

30.3.4 Time alarm

The Time Alarm register (TAR), SR[TAF], and IER[TAIE] allow the RTC to generate an interrupt at a predefined time. The 32-bit TAR is compared with the 32-bit Time Seconds register (TSR) each time it increments. SR[TAF] will set when TAR equals TSR and TSR increments.

SR[TAF] is cleared by writing TAR. This will usually be the next alarm value, although writing a value that is less than TSR, such as 0, will prevent SR[TAF] from setting again. SR[TAF] cannot otherwise be disabled, although the interrupt it generates is enabled or disabled by IER[TAIE].

30.3.5 Update mode

The Update Mode field in the Control register (CR[UM]) configures software write access to the Time Counter Enable (SR[TCE]) field. When CR[UM] is clear, SR[TCE] can be written only when LR[SRL] is set. When CR[UM] is set, SR[TCE] can also be written when SR[TCE] is clear or when SR[TIF] or SR[TOF] are set. This allows the time seconds and prescaler registers to be initialized whenever time is invalidated, while preventing the time seconds and prescaler registers from being changed on the fly. When LR[SRL] is set, CR[UM] has no effect on SR[TCE].

30.3.6 Register lock

The Lock register (LR) can be used to block write accesses to certain registers until the next POR or software reset. Locking the Control register (CR) will disable the software reset. Locking LR will block future updates to LR.

Write accesses to a locked register are ignored and do not generate a bus error.

30.3.7 Interrupt

The RTC interrupt is asserted whenever a status flag and the corresponding interrupt enable bit are both set. It is always asserted on POR, and software reset. The RTC interrupt is enabled at the chip level by enabling the chip-specific RTC clock gate control bit. The RTC interrupt can be used to wakeup the chip from any low-power mode.

The optional RTC seconds interrupt is an edge-sensitive interrupt with a dedicated interrupt vector that is generated once a second and requires no software overhead (there is no corresponding status flag to clear). It is enabled in the RTC by the time seconds interrupt enable bit and enabled at the chip level by setting the chip-specific RTC clock gate control bit. This interrupt is optional and may not be implemented on all devices.

Chapter 31
Serial Peripheral Interface (SPI)

31.1 Introduction

NOTE

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

The serial peripheral interface (SPI) module provides for full-duplex, synchronous, serial communication between the MCU and peripheral devices. These peripheral devices can include other microcontrollers, analog-to-digital converters, shift registers, sensors, and memories, among others.

The SPI runs at a baud rate up to the SPI module clock divided by two in master mode and up to the SPI module clock divided by four in slave mode. Software can poll the status flags, or SPI operation can be interrupt driven.

NOTE

For the actual maximum SPI baud rate, refer to the Chip Configuration details and to the device’s Data Sheet.

The SPI also supports a data length of 8 or 16 bits and includes a hardware match feature for the receive data buffer.

The SPI includes an internal DMA interface to support continuous SPI transmission through an on-chip DMA controller instead of through the CPU. This feature decreases CPU loading, allowing CPU time to be used for other work.

31.1.1 Features

The SPI includes these distinctive features:

- Master mode or slave mode operation

- Full-duplex or single-wire bidirectional mode
- Programmable transmit bit rate
- Double-buffered transmit and receive data register
- Serial clock phase and polarity options
- Slave select output
- Mode fault error flag with CPU interrupt capability
- Control of SPI operation during wait mode
- Selectable MSB-first or LSB-first shifting
- Programmable 8- or 16-bit data transmission length
- Receive data buffer hardware match feature
- 64-bit FIFO mode for high speed/large amounts of data transfers
- Support transmission of both Transmit and Receive by DMA

31.1.2 Modes of operation

The SPI functions in the following three modes.

- Run mode

  This is the basic mode of operation.

- Wait mode

  SPI operation in Wait mode is a configurable low power mode, controlled by the SPISWAI bit located in the SPIx_C2 register. In Wait mode, if C2[SPISWAI] is clear, the SPI operates like in Run mode. If C2[SPISWAI] is set, the SPI goes into a power conservative state, with the SPI clock generation turned off. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU enters Run mode. If the SPI is configured as a slave, reception and transmission of a byte continues, so that the slave stays synchronized to the master.

- Stop mode

To reduce power consumption, the SPI is inactive in stop modes where the peripheral bus clock is stopped but internal logic states are retained. If the SPI is configured as a master, any transmission in progress stops, but is resumed after the CPU enters run mode. If the SPI is configured as a slave, reception and transmission of a data continues, so that the slave stays synchronized to the master.

The SPI is completely disabled in Stop modes where the peripheral bus clock is stopped and internal logic states are not retained. When the CPU wakes from these Stop modes, all SPI register content is reset.

Detailed descriptions of operating modes appear in Low-power mode options.

31.1.3 Block diagrams

This section includes block diagrams showing SPI system connections, the internal organization of the SPI module, and the SPI clock dividers that control the master mode bit rate.

31.1.3.1 SPI system block diagram

The following figure shows the SPI modules of two MCUs connected in a master-slave arrangement. The master device initiates all SPI data transfers. During a transfer, the master shifts data out (on the MOSI pin) to the slave while simultaneously shifting data in (on the MISO pin) from the slave. The transfer effectively exchanges the data that was in the SPI shift registers of the two SPI systems. The SPSCK signal is a clock output from the master and an input to the slave. The slave device must be selected by a low level on the slave select input (SS pin). In this system, the master device has configured its SS pin as an optional slave select output.

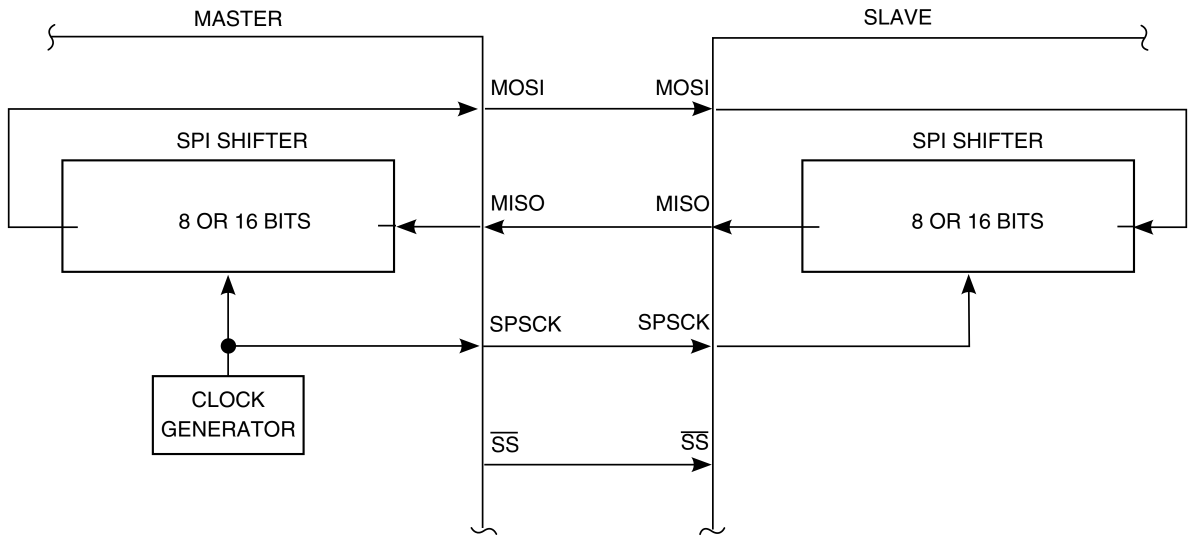


Figure 31-1. SPI system connections

31.1.3.2 SPI module block diagram

The following is a block diagram of the SPI module. The central element of the SPI is the SPI shift register. Data is written to the double-buffered transmitter (write to SPIx_DH:SPIx_DL) and gets transferred to the SPI Shift Register at the start of a data transfer. After shifting in 8 bits or 16 bits (as determined by the SPIMODE bit) of data, the data is transferred into the double-buffered receiver where it can be read from SPIx_DH:SPIx_DL. Pin multiplexing logic controls connections between MCU pins and the SPI module.

When the FIFO feature is supported: Additionally there is an 8-byte receive FIFO and an 8-byte transmit FIFO that (once enabled) provide features to allow fewer CPU interrupts to occur when transmitting/receiving high volume/high speed data. When FIFO mode is enabled, the SPI can still function in either 8-bit or 16-bit mode (as per SPIMODE bit) and three additional flags help monitor the FIFO status. Two of these flags can provide CPU interrupts.

When the SPI is configured as a master, the clock output is routed to the SPSCK pin, the shifter output is routed to MOSI, and the shifter input is routed from the MISO pin.

When the SPI is configured as a slave, the SPSCK pin is routed to the clock input of the SPI, the shifter output is routed to MISO, and the shifter input is routed from the MOSI pin.

In the external SPI system, simply connect all SPSCK pins to each other, all MISO pins together, and all MOSI pins together. Peripheral devices often use slightly different names for these pins.

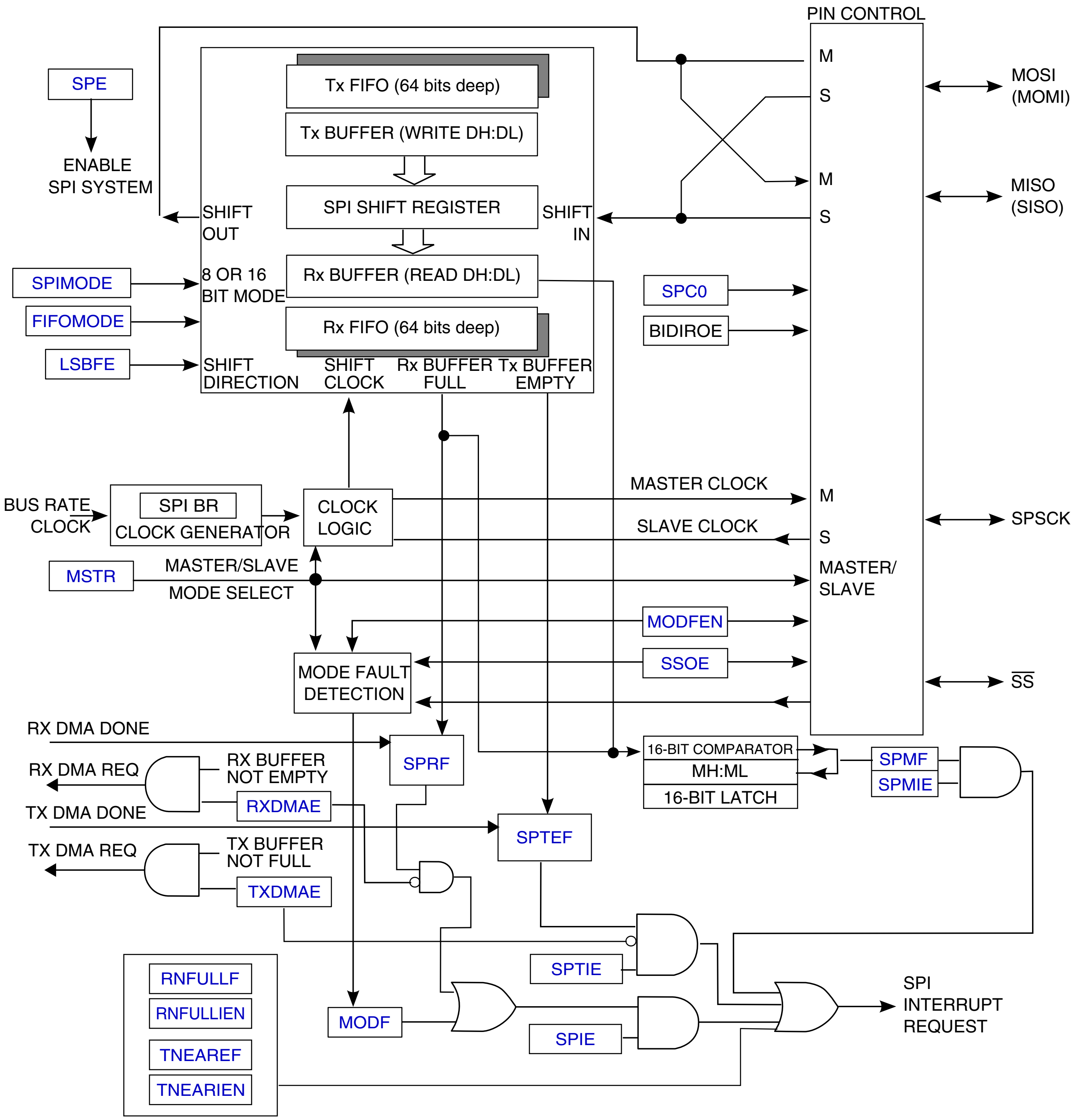


Figure 31-2. SPI Module Block Diagram with FIFO

31.2 External signal description

The SPI optionally shares four port pins. The function of these pins depends on the settings of SPI control bits. When the SPI is disabled (SPE = 0), these four pins revert to other functions that are not controlled by the SPI (based on chip configuration).

31.2.1 SPSCK — SPI Serial Clock

When the SPI is enabled as a slave, this pin is the serial clock input. When the SPI is enabled as a master, this pin is the serial clock output.

31.2.2 MOSI — Master Data Out, Slave Data In

When the SPI is enabled as a master and SPI pin control zero (SPC0) is 0 (not bidirectional mode), this pin is the serial data output. When the SPI is enabled as a slave and SPC0 is 0, this pin is the serial data input. If SPC0 is 1 to select single-wire bidirectional mode, and master mode is selected, this pin becomes the bidirectional data I/O pin (MOMI). Also, the bidirectional mode output enable bit determines whether the pin acts as an input (BIDIROE is 0) or an output (BIDIROE is 1). If SPC0 is 1 and slave mode is selected, this pin is not used by the SPI and reverts to other functions (based on chip configuration).

31.2.3 MISO — Master Data In, Slave Data Out

When the SPI is enabled as a master and SPI pin control zero (SPC0) is 0 (not bidirectional mode), this pin is the serial data input. When the SPI is enabled as a slave and SPC0 is 0, this pin is the serial data output. If SPC0 is 1 to select single-wire bidirectional mode, and slave mode is selected, this pin becomes the bidirectional data I/O pin (SISO), and the bidirectional mode output enable bit determines whether the pin acts as an input (BIDIROE is 0) or an output (BIDIROE is 1). If SPC0 is 1 and master mode is selected, this pin is not used by the SPI and reverts to other functions (based on chip configuration).

31.2.4 $\overline{\text{SS}}$ — Slave Select

When the SPI is enabled as a slave, this pin is the low-true slave select input. When the SPI is enabled as a master and mode fault enable is off (MODFEN is 0), this pin is not used by the SPI and reverts to other functions (based on chip configuration). When the SPI is enabled as a master and MODFEN is 1, the slave select output enable bit determines whether this pin acts as the mode fault input (SSOE is 0) or as the slave select output (SSOE is 1).

31.3 Memory map/register definition

The SPI has 8-bit registers to select SPI options, to control baud rate, to report SPI status, to hold an SPI data match value, and for transmit/receive data.

SPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_6000	SPI Status Register (SPI0_S)	8	R	20h	31.3.1/571
4007_6001	SPI Baud Rate Register (SPI0_BR)	8	R/W	00h	31.3.2/575
4007_6002	SPI Control Register 2 (SPI0_C2)	8	R/W	00h	31.3.3/576
4007_6003	SPI Control Register 1 (SPI0_C1)	8	R/W	04h	31.3.4/577
4007_6004	SPI Match Register low (SPI0_ML)	8	R/W	00h	31.3.5/579
4007_6005	SPI match register high (SPI0_MH)	8	R/W	00h	31.3.6/580
4007_6006	SPI Data Register low (SPI0_DL)	8	R/W	00h	31.3.7/580
4007_6007	SPI data register high (SPI0_DH)	8	R/W	00h	31.3.8/581
4007_600A	SPI clear interrupt register (SPI0_CI)	8	R/W	00h	31.3.9/581
4007_600B	SPI control register 3 (SPI0_C3)	8	R/W	00h	31.3.10/ 583
4007_7000	SPI Status Register (SPI1_S)	8	R	20h	31.3.1/571
4007_7001	SPI Baud Rate Register (SPI1_BR)	8	R/W	00h	31.3.2/575
4007_7002	SPI Control Register 2 (SPI1_C2)	8	R/W	00h	31.3.3/576
4007_7003	SPI Control Register 1 (SPI1_C1)	8	R/W	04h	31.3.4/577
4007_7004	SPI Match Register low (SPI1_ML)	8	R/W	00h	31.3.5/579
4007_7005	SPI match register high (SPI1_MH)	8	R/W	00h	31.3.6/580
4007_7006	SPI Data Register low (SPI1_DL)	8	R/W	00h	31.3.7/580
4007_7007	SPI data register high (SPI1_DH)	8	R/W	00h	31.3.8/581
4007_700A	SPI clear interrupt register (SPI1_CI)	8	R/W	00h	31.3.9/581
4007_700B	SPI control register 3 (SPI1_C3)	8	R/W	00h	31.3.10/ 583

31.3.1 SPI Status Register (SPIx_S)

This register contains read-only status bits. Writes have no meaning or effect.

When the FIFO is supported and enabled (FIFOMODE is 1): This register has four flags that provide mechanisms to support an 8-byte FIFO mode: RNFULLF, TNEARF, TXFULLF, and RFIFOEF. When the SPI is in 8-byte FIFO mode, the function of SPRF and SPTEF differs slightly from their function in the normal buffered modes, mainly regarding how these flags are cleared by the amount available in the transmit and receive FIFOs.

- The RNFULLF and TNEAREF help improve the efficiency of FIFO operation when transfering large amounts of data. These flags provide a "watermark" feature of the FIFOs to allow continuous transmissions of data when running at high speed.
- The RNFULLF can generate an interrupt if the RNFULLIEN bit in the C3 register is set, which allows the CPU to start emptying the receive FIFO without delaying the reception of subsequent bytes. The user can also determine if all data in the receive FIFO has been read by monitoring the RFIFOEF.
- The TNEAREF can generate an interrupt if the TNEARIEN bit in the C3 register is set, which allows the CPU to start filling the transmit FIFO before it is empty and thus to prevent breaks in SPI transmission.

NOTE

At an initial POR, the values of TNEAREF and RFIFOEF are 0. However, the status (S) register and both TX and RX FIFOs are reset due to a change of SPIMODE, FIFOMODE or SPE. If this type of reset occurs and FIFOMODE is 0, TNEAREF and RFIFOEF continue to reset to 0. If this type of reset occurs and FIFOMODE is 1, TNEAREF and RFIFOEF reset to 1.

Address: Base address + 0h offset

Bit	7	6	5	4	3	2	1	0
Read	SPRF	SPMF	SPTEF	MODF	RNFULLF	TNEAREF	TXFULLF	RFIFOEF
Write								
Reset	0	0	1	0	0	0	0	0

SPIx_S field descriptions

Field	Description
7 SPRF	SPI Read Buffer Full Flag (when FIFO is not supported or not enabled) or SPI read FIFO FULL flag (when FIFO is supported and enabled) When the FIFO is not supported or not enabled (FIFOMODE is not present or is 0): SPRF is set at the completion of an SPI transfer to indicate that received data may be read from the SPI data (DH:DL) register. When the receive DMA request is disabled (RXDMAE is 0), SPRF is cleared by reading SPRF while it is set and then reading the SPI data register. When the receive DMA request is enabled (RXDMAE is 1), SPRF is automatically cleared when the DMA transfer for the receive DMA request is completed (RX DMA Done is asserted). When FIFOMODE is 1: This bit indicates the status of the read FIFO when FIFOMODE is enabled. The SPRF is set when the read FIFO has received 64 bits (4 words or 8 bytes) of data from the shifter and there have been no CPU reads of the SPI data (DH:DL) register. When the receive DMA request is disabled (RXDMAE is 0), SPRF is cleared by reading the SPI data register, resulting in the FIFO no longer being full, assuming another SPI message is not received. When the receive DMA request is enabled (RXDMAE is 1), SPRF is automatically cleared when the first DMA transfer for the receive DMA request is completed (RX DMA Done is asserted).

Table continues on the next page...

SPIx_S field descriptions (continued)

Field	Description
	0 No data available in the receive data buffer (when FIFOMODE is not present or is 0) or Read FIFO is not full (when FIFOMODE is 1) 1 Data available in the receive data buffer (when FIFOMODE is not present or is 0) or Read FIFO is full (when FIFOMODE is 1)
6 SPMF	SPI Match Flag SPMF is set after SPRF is 1 when the value in the receive data buffer matches the value in the MH:ML registers. To clear the flag, read SPMF when it is set and then write a 1 to it. 0 Value in the receive data buffer does not match the value in the MH:ML registers 1 Value in the receive data buffer matches the value in the MH:ML registers
5 SPTEF	SPI Transmit Buffer Empty Flag (when FIFO is not supported or not enabled) or SPI transmit FIFO empty flag (when FIFO is supported and enabled) When the FIFO is not supported or not enabled (FIFOMODE is not present or is 0): This bit is set when the transmit data buffer is empty. When the transmit DMA request is disabled (TXDMAE is 0), SPTEF is cleared by reading the S register with SPTEF set and then writing a data value to the transmit buffer at DH:DL. The S register must be read with SPTEF set to 1 before writing data to the DH:DL register; otherwise, the DH:DL write is ignored. When the transmit DMA request is enabled (TXDMAE is 1), SPTEF is automatically cleared when the DMA transfer for the transmit DMA request is completed (TX DMA Done is asserted). SPTEF is automatically set when all data from the transmit buffer transfers into the transmit shift register. For an idle SPI, data written to DH:DL is transferred to the shifter almost immediately so that SPTEF is set within two bus cycles, allowing a second set of data to be queued into the transmit buffer. After completion of the transfer of the data in the shift register, the queued data from the transmit buffer automatically moves to the shifter, and SPTEF is set to indicate that room exists for new data in the transmit buffer. If no new data is waiting in the transmit buffer, SPTEF simply remains set and no data moves from the buffer to the shifter. When the FIFO is supported and enabled (FIFOMODE is 1): This bit provides the status of the FIFO rather than an 8-bit or a 16-bit buffer. This bit is set when the transmit FIFO is empty. When the transmit DMA request is disabled (TXDMAE is 0), SPTEF is cleared by writing a data value to the transmit FIFO at DH:DL. When the transmit DMA request is enabled (TXDMAE is 1), SPTEF is automatically cleared when the DMA transfer for the transmit DMA request is completed (TX DMA Done is asserted). SPTEF is automatically set when all data from the transmit FIFO transfers into the transmit shift register. For an idle SPI, data written to the DH:DL register is transferred to the shifter almost immediately, so that SPTEF is set within two bus cycles. A second write of data to the DH:DL register clears this SPTEF flag. After completion of the transfer of the data in the shift register, the queued data from the transmit FIFO automatically moves to the shifter, and SPTEF will be set only when all data written to the transmit FIFO has been transfered to the shifter. If no new data is waiting in the transmit FIFO, SPTEF simply remains set and no data moves from the buffer to the shifter. 0 SPI transmit buffer not empty (when FIFOMODE is not present or is 0) or SPI FIFO not empty (when FIFOMODE is 1) 1 SPI transmit buffer empty (when FIFOMODE is not present or is 0) or SPI FIFO empty (when FIFOMODE is 1)
4 MODF	Master Mode Fault Flag MODF is set if the SPI is configured as a master and the slave select input goes low, indicating some other SPI device is also configured as a master. The $\overline{SS}$ pin acts as a mode fault error input only when C1[MSTR] is 1, C2[MODFEN] is 1, and C1[SSOE] is 0; otherwise, MODF will never be set. MODF is cleared by reading MODF while it is 1 and then writing to the SPI Control Register 1 (C1). 0 No mode fault error 1 Mode fault error detected

Table continues on the next page...

SPIx_S field descriptions (continued)

Field	Description
3 RNFULLF	Receive FIFO nearly full flag This flag is set when more than three 16-bit words or six 8-bit bytes of data remain in the receive FIFO, provided C3[RNFULLF_MARK] is 0, or when more than two 16-bit words or four 8-bit bytes of data remain in the receive FIFO, provided C3[RNFULLF_MARK] is 1. It has no function if FIFOMODE is not present or is 0. 0 Receive FIFO has received less than 48 bits (when C3[RNFULLF_MARK] is 0) or less than 32 bits (when C3[RNFULLF_MARK] is 1) 1 Receive FIFO has received data of an amount equal to or greater than 48 bits (when C3[RNFULLF_MARK] is 0) or 32 bits (when C3[RNFULLF_MARK] is 1)
2 TNEAREF	Transmit FIFO nearly empty flag This flag is set when only one 16-bit word or two 8-bit bytes of data remain in the transmit FIFO, provided C3[TNEAREF_MARK] is 0, or when only two 16-bit words or four 8-bit bytes of data remain in the transmit FIFO, provided C3[TNEAREF_MARK] is 1. If FIFOMODE is not enabled, ignore this bit. **NOTE:** At an initial POR, the values of TNEAREF and RFIFOEF are 0. However, the status (S) register and both TX and RX FIFOs are reset due to a change of SPIMODE, FIFOMODE or SPE. If this type of reset occurs and FIFOMODE is 0, TNEAREF and RFIFOEF continue to reset to 0. If this type of reset occurs and FIFOMODE is 1, TNEAREF and RFIFOEF reset to 1. 0 Transmit FIFO has more than 16 bits (when C3[TNEAREF_MARK] is 0) or more than 32 bits (when C3[TNEAREF_MARK] is 1) remaining to transmit 1 Transmit FIFO has an amount of data equal to or less than 16 bits (when C3[TNEAREF_MARK] is 0) or 32 bits (when C3[TNEAREF_MARK] is 1) remaining to transmit
1 TXFULLF	Transmit FIFO full flag This bit indicates the status of the transmit FIFO when FIFOMODE is enabled. This flag is set when there are 8 bytes in the transmit FIFO. If FIFOMODE is not enabled, ignore this bit. When FIFOMODE and DMA are both enabled, the inverted TXFULLF is used to trigger a DMA transfer. So when the transmit FIFO is not full, the DMA request is active, and remains active until the FIFO is full. 0 Transmit FIFO has less than 8 bytes 1 Transmit FIFO has 8 bytes of data
0 RFIFOEF	SPI read FIFO empty flag This bit indicates the status of the read FIFO when FIFOMODE is enabled. If FIFOMODE is not enabled, ignore this bit. When FIFOMODE and DMA are both enabled, the inverted RXIFOEF is used to trigger a DMA transfer. So when the receive FIFO is not empty, the DMA request is active, and remains active until the FIFO is empty. **NOTE:** At an initial POR, the values of TNEAREF and RFIFOEF are 0. However, the status (S) register and both TX and RX FIFOs are reset due to a change of SPIMODE, FIFOMODE or SPE. If this type of reset occurs and FIFOMODE is 0, TNEAREF and RFIFOEF continue to reset to 0. If this type of reset occurs and FIFOMODE is 1, TNEAREF and RFIFOEF reset to 1. 0 Read FIFO has data. Reads of the DH:DL registers in 16-bit mode or the DL register in 8-bit mode will empty the read FIFO. 1 Read FIFO is empty.

31.3.2 SPI Baud Rate Register (SPIx_BR)

Use this register to set the prescaler and bit rate divisor for an SPI master. This register may be read or written at any time.

Address: Base address + 1h offset

Bit	7	6	5	4	3	2	1	0
Read	0	SPPR[2:0]			SPR[3:0]			
Write								
Reset	0	0	0	0	0	0	0	0

SPIx_BR field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 SPPR[2:0]	SPI Baud Rate Prescale Divisor This 3-bit field selects one of eight divisors for the SPI baud rate prescaler. The input to this prescaler is the SPI module clock. The output of this prescaler drives the input of the SPI baud rate divider. Refer to the description of “SPI Baud Rate Generation” for details. 000 Baud rate prescaler divisor is 1. 001 Baud rate prescaler divisor is 2. 010 Baud rate prescaler divisor is 3. 011 Baud rate prescaler divisor is 4. 100 Baud rate prescaler divisor is 5. 101 Baud rate prescaler divisor is 6. 110 Baud rate prescaler divisor is 7. 111 Baud rate prescaler divisor is 8.
3–0 SPR[3:0]	SPI Baud Rate Divisor This 4-bit field selects one of nine divisors for the SPI baud rate divider. The input to this divider comes from the SPI baud rate prescaler. Refer to the description of “SPI Baud Rate Generation” for details. 0000 Baud rate divisor is 2. 0001 Baud rate divisor is 4. 0010 Baud rate divisor is 8. 0011 Baud rate divisor is 16. 0100 Baud rate divisor is 32. 0101 Baud rate divisor is 64. 0110 Baud rate divisor is 128. 0111 Baud rate divisor is 256. 1000 Baud rate divisor is 512. All others Reserved

31.3.3 SPI Control Register 2 (SPIx_C2)

This read/write register is used to control optional features of the SPI system.

Address: Base address + 2h offset

Bit	7	6	5	4	3	2	1	0
Read Write	SPMIE	SPIMODE	TXDMAE	MODFEN	BIDIROE	RXDMAE	SPISWAI	SPC0
Reset	0	0	0	0	0	0	0	0

SPIx_C2 field descriptions

Field	Description
7 SPMIE	SPI Match Interrupt Enable This is the interrupt enable bit for the SPI receive data buffer hardware match (SPMF) function. 0 Interrupts from SPMF inhibited (use polling) 1 When SPMF is 1, requests a hardware interrupt
6 SPIMODE	SPI 8-bit or 16-bit mode This bit allows the user to select either an 8-bit or 16-bit SPI data transmission length. In master mode, a change of this bit aborts a transmission in progress, forces the SPI system into an idle state, and resets all status bits in the S register. Refer to the description of “Data Transmission Length” for details. 0 8-bit SPI shift register, match register, and buffers 1 16-bit SPI shift register, match register, and buffers
5 TXDMAE	Transmit DMA enable This bit enables a transmit DMA request. When this bit is set to 1, a transmit DMA request is asserted when both SPTEF and SPE are set, and the interrupt from SPTEF is disabled. 0 DMA request for transmit is disabled and interrupt from SPTEF is allowed 1 DMA request for transmit is enabled and interrupt from SPTEF is disabled
4 MODFEN	Master Mode-Fault Function Enable When the SPI is configured for slave mode, this bit has no meaning or effect. (The $\overline{SS}$ pin is the slave select input.) In master mode, this bit determines how the $\overline{SS}$ pin is used. For details, refer to the description of the SSOE bit in the C1 register. 0 Mode fault function disabled, master $\overline{SS}$ pin reverts to general-purpose I/O not controlled by SPI 1 Mode fault function enabled, master $\overline{SS}$ pin acts as the mode fault input or the slave select output
3 BIDIROE	Bidirectional Mode Output Enable When bidirectional mode is enabled because SPI pin control 0 (SPC0) is set to 1, BIDIROE determines whether the SPI data output driver is enabled to the single bidirectional SPI I/O pin. Depending on whether the SPI is configured as a master or a slave, it uses the MOSI (MOMI) or MISO (SISO) pin, respectively, as the single SPI data I/O pin. When SPC0 is 0, BIDIROE has no meaning or effect. 0 Output driver disabled so SPI data I/O pin acts as an input 1 SPI I/O pin enabled as an output

Table continues on the next page...

SPIx_C2 field descriptions (continued)

Field	Description
2 RXDMAE	Receive DMA enable This is the enable bit for a receive DMA request. When this bit is set to 1, a receive DMA request is asserted when both SPRF and SPE are set, and the interrupt from SPRF is disabled. 0 DMA request for receive is disabled and interrupt from SPRF is allowed 1 DMA request for receive is enabled and interrupt from SPRF is disabled
1 SPISWAI	SPI Stop in Wait Mode This bit is used for power conservation while the device is in Wait mode. 0 SPI clocks continue to operate in Wait mode. 1 SPI clocks stop when the MCU enters Wait mode.
0 SPC0	SPI Pin Control 0 Enables bidirectional pin configurations. 0 SPI uses separate pins for data input and data output (pin mode is normal). In master mode of operation: MISO is master in and MOSI is master out. In slave mode of operation: MISO is slave out and MOSI is slave in. 1 SPI configured for single-wire bidirectional operation (pin mode is bidirectional). In master mode of operation: MISO is not used by SPI; MOSI is master in when BIDIROE is 0 or master I/O when BIDIROE is 1. In slave mode of operation: MISO is slave in when BIDIROE is 0 or slave I/O when BIDIROE is 1; MOSI is not used by SPI.

31.3.4 SPI Control Register 1 (SPIx_C1)

This read/write register includes the SPI enable control, interrupt enables, and configuration options.

Address: Base address + 3h offset

Bit	7	6	5	4	3	2	1	0
Read Write	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset	0	0	0	0	0	1	0	0

SPIx_C1 field descriptions

Field	Description
7 SPIE	SPI Interrupt Enable: for SPRF and MODF (when FIFO is not supported or not enabled) or for read FIFO (when FIFO is supported and enabled) When the FIFO is not supported or not enabled (FIFOMODE is not present or is 0): Enables the interrupt for SPI receive buffer full (SPRF) and mode fault (MODF) events.

Table continues on the next page...

SPIx_C1 field descriptions (continued)

Field	Description
	When the FIFO is supported and enabled (FIFOMODE is 1): This bit enables the SPI to interrupt the CPU when the receive FIFO is full. An interrupt occurs when the SPRF bit is set or the MODF bit is set. 0 Interrupts from SPRF and MODF are inhibited—use polling (when FIFOMODE is not present or is 0) or Read FIFO Full Interrupts are disabled (when FIFOMODE is 1) 1 Request a hardware interrupt when SPRF or MODF is 1 (when FIFOMODE is not present or is 0) or Read FIFO Full Interrupts are enabled (when FIFOMODE is 1)
6 SPE	SPI System Enable Enables the SPI system and dedicates the SPI port pins to SPI system functions. If SPE is cleared, the SPI is disabled and forced into an idle state, and all status bits in the S register are reset. 0 SPI system inactive 1 SPI system enabled
5 SPTIE	SPI Transmit Interrupt Enable When the FIFO is not supported or not enabled (FIFOMODE is not present or is 0): This is the interrupt enable bit for SPI transmit buffer empty (SPTEF). An interrupt occurs when the SPI transmit buffer is empty (SPTEF is set). When the FIFO is supported and enabled (FIFOMODE is 1): This is the interrupt enable bit for SPI transmit FIFO empty (SPTEF). An interrupt occurs when the SPI transmit FIFO is empty (SPTEF is set). 0 Interrupts from SPTEF inhibited (use polling) 1 When SPTEF is 1, hardware interrupt requested
4 MSTR	Master/Slave Mode Select Selects master or slave mode operation. 0 SPI module configured as a slave SPI device 1 SPI module configured as a master SPI device
3 CPOL	Clock Polarity Selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. This bit effectively places an inverter in series with the clock signal either from a master SPI device or to a slave SPI device. Refer to the description of “SPI Clock Formats” for details. 0 Active-high SPI clock (idles low) 1 Active-low SPI clock (idles high)
2 CPHA	Clock Phase Selects one of two clock formats for different kinds of synchronous serial peripheral devices. Refer to the description of “SPI Clock Formats” for details. 0 First edge on SPSCK occurs at the middle of the first cycle of a data transfer. 1 First edge on SPSCK occurs at the start of the first cycle of a data transfer.
1 SSOE	Slave Select Output Enable This bit is used in combination with the Mode Fault Enable (MODFEN) field in the C2 register and the Master/Slave (MSTR) control bit to determine the function of the $\overline{SS}$ pin. 0 When C2[MODFEN] is 0: In master mode, $\overline{SS}$ pin function is general-purpose I/O (not SPI). In slave mode, $\overline{SS}$ pin function is slave select input.

Table continues on the next page...

SPIx_C1 field descriptions (continued)

Field	Description
	When C2[MODFEN] is 1: In master mode, $\overline{SS}$ pin function is $\overline{SS}$ input for mode fault. In slave mode, $\overline{SS}$ pin function is slave select input. 1 When C2[MODFEN] is 0: In master mode, $\overline{SS}$ pin function is general-purpose I/O (not SPI). In slave mode, $\overline{SS}$ pin function is slave select input. When C2[MODFEN] is 1: In master mode, $\overline{SS}$ pin function is automatic $\overline{SS}$ output. In slave mode: $\overline{SS}$ pin function is slave select input.
0 LSBFE	LSB First (shifter direction) This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in bit 7 (or bit 15 in 16-bit mode). 0 SPI serial data transfers start with the most significant bit. 1 SPI serial data transfers start with the least significant bit.

31.3.5 SPI Match Register low (SPIx_ML)

This register, together with the MH register, contains the hardware compare value. When the value received in the SPI receive data buffer equals this hardware compare value, the SPI Match Flag in the S register (S[SPMF]) sets.

In 8-bit mode, only the ML register is available. Reads of the MH register return all zeros. Writes to the MH register are ignored.

In 16-bit mode, reading either byte (the MH or ML register) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. Writing to either byte (the MH or ML register) latches the value into a buffer. When both bytes have been written, they are transferred as a coherent value into the SPI match registers.

Address: Base address + 4h offset

Bit	7	6	5	4	3	2	1	0
Read Write	Bits[7:0]							
Reset	0	0	0	0	0	0	0	0

SPIx_ML field descriptions

Field	Description
7–0 Bits[7:0]	Hardware compare value (low byte)

31.3.6 SPI match register high (SPIx_MH)

Refer to the description of the ML register.

Address: Base address + 5h offset

Bit	7	6	5	4	3	2	1	0
Read / Write	Bits[15:8]							
Reset	0	0	0	0	0	0	0	0

SPIx_MH field descriptions

Field	Description
7–0 Bits[15:8]	Hardware compare value (high byte)

31.3.7 SPI Data Register low (SPIx_DL)

This register, together with the DH register, is both the input and output register for SPI data. A write to the registers writes to the transmit data buffer, allowing data to be queued and transmitted.

When the SPI is configured as a master, data queued in the transmit data buffer is transmitted immediately after the previous transmission has completed.

The SPTEF bit in the S register indicates when the transmit data buffer is ready to accept new data. When the transmit DMA request is disabled (TXDMAE is 0): The S register must be read when S[SPTEF] is set before writing to the SPI data registers; otherwise, the write is ignored. When the transmit DMA request is enabled (TXDMAE is 1) when S[SPTEF] is set, the SPI data registers can be written automatically by DMA without reading the S register first.

Data may be read from the SPI data registers any time after S[SPRF] is set and before another transfer is finished. Failure to read the data out of the receive data buffer before a new transfer ends causes a receive overrun condition, and the data from the new transfer is lost. The new data is lost because the receive buffer still held the previous character and was not ready to accept the new data. There is no indication for a receive overrun condition, so the application system designer must ensure that previous data has been read from the receive buffer before a new transfer is initiated.

In 8-bit mode, only the DL register is available. Reads of the DH register return all zeros. Writes to the DH register are ignored.

In 16-bit mode, reading either byte (the DH or DL register) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. Writing to either byte (the DH or DL register) latches the value into a buffer. When both bytes have been written, they are transferred as a coherent 16-bit value into the transmit data buffer.

Address: Base address + 6h offset

Bit	7	6	5	4	3	2	1	0
Read Write	Bits[7:0]							
Reset	0	0	0	0	0	0	0	0

SPIx_DL field descriptions

Field	Description
7–0 Bits[7:0]	Data (low byte)

31.3.8 SPI data register high (SPIx_DH)

Refer to the description of the DL register.

Address: Base address + 7h offset

Bit	7	6	5	4	3	2	1	0
Read Write	Bits[15:8]							
Reset	0	0	0	0	0	0	0	0

SPIx_DH field descriptions

Field	Description
7–0 Bits[15:8]	Data (high byte)

31.3.9 SPI clear interrupt register (SPIx_CI)

This register applies only for an instance of the SPI module that supports the FIFO feature.

The register has four bits dedicated to clearing the interrupts. Writing 1 to these bits clears the corresponding interrupts if the INTCLR bit in the C3 register is 1. Reading these bits always returns 0.

This register also has two read-only bits to indicate the transmit FIFO and receive FIFO overrun conditions. When the receive FIFO is full and data is received, RXFOF is set. Similarily, when the transmit FIFO is full and a write to the data register occurs, TXFOF is set. These flags are cleared when the CI register is read while the flags are set.

The register has two more read-only bits to indicate the error flags. These flags are set when, due to some spurious reason, entries in the FIFO total more than 64 bits of data. At this point, all the flags in the status register are reset, and entries in the FIFO are flushed with the corresponding error flags set. These flags are cleared when the CI register is read while the flags are set.

Address: Base address + Ah offset

Bit	7	6	5	4	3	2	1	0
Read	TXFERR	RXFERR	TXFOF	RXFOF	0	0	0	0
Write					TNEAREFCI	RNFULLFCI	SPTEFCI	SPRFCI
Reset	0	0	0	0	0	0	0	0

SPIx_CI field descriptions

Field	Description
7 TXFERR	Transmit FIFO error flag This flag indicates that a transmit FIFO error occurred because entries in the FIFO total more than 64 bits of data. 0 No transmit FIFO error occurred 1 A transmit FIFO error occurred
6 RXFERR	Receive FIFO error flag This flag indicates that a receive FIFO error occurred because entries in the FIFO total more than 64 bits of data. 0 No receive FIFO error occurred 1 A receive FIFO error occurred
5 TXFOF	Transmit FIFO overflow flag This flag indicates that a transmit FIFO overflow condition has occurred. 0 Transmit FIFO overflow condition has not occurred 1 Transmit FIFO overflow condition occurred
4 RXFOF	Receive FIFO overflow flag This flag indicates that a receive FIFO overflow condition has occurred. 0 Receive FIFO overflow condition has not occurred 1 Receive FIFO overflow condition occurred
3 TNEAREFCI	Transmit FIFO nearly empty flag clear interrupt Writing 1 to this bit clears the TNEAREF interrupt provided that C3[3] is set.
2 RNFULLFCI	Receive FIFO nearly full flag clear interrupt Writing 1 to this bit clears the RNFULLF interrupt provided that C3[3] is set.

Table continues on the next page...

SPIx_CI field descriptions (continued)

Field	Description
1 SPTEFCI	Transmit FIFO empty flag clear interrupt Writing 1 to this bit clears the SPTEF interrupt provided that C3[3] is set.
0 SPRFCI	Receive FIFO full flag clear interrupt Writing 1 to this bit clears the SPRF interrupt provided that C3[3] is set.

31.3.10 SPI control register 3 (SPIx_C3)

This register introduces a 64-bit FIFO function on both transmit and receive buffers. It applies only for an instance of the SPI module that supports the FIFO feature.

FIFO mode is enabled by setting the FIFOMODE bit to 1. A write to this register occurs only when it sets the FIFOMODE bit to 1.

Using this FIFO feature allows the SPI to provide high speed transfers of large amounts of data without consuming large amounts of the CPU bandwidth.

Enabling this FIFO function affects the behavior of some of the read/write buffer flags in the S register as follows:

- When the receive FIFO has data in it, S[RFIFOEF] is 0. As a result:
  - If C2[RXDMAE] is 1, RFIFOEF_b generates a receive DMA request. The DMA request remains active until RFIFOEF is set to 1, indicating the receive buffer is empty.
- If C2[RXDMAE] is 0 and C1[SPIE] is 1, SPRF interrupts the CPU.
- When the transmit FIFO is not full, S[TXFULLF] is 0. As a result:
  - If C2[TXDMAE] is 1, TXFULLF_b generates a transmit DMA request. The DMA request remains active until TXFULLF is set to 1, indicating the transmit FIFO is full.
- If C2[TXDMAE] is 0 and C1[SPTIE] is 1, SPTEF interrupts the CPU.

Two interrupt enable bits, TNEARIEN and RNFULLIEN, provide CPU interrupts based on the "watermark" feature of the TNEARF and RNFULLF flags of the S register.

Address: Base address + Bh offset

Bit	7	6	5	4	3	2	1	0
Read	0		TNEAREF_MARK	RNFULLF_MARK	INTCLR	TNEARIEN	RNFULLIEN	FIFOMODE
Write								
Reset	0	0	0	0	0	0	0	0

SPIx_C3 field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 TNEAREF_ MARK	Transmit FIFO nearly empty watermark This bit selects the mark after which the TNEAREF flag is asserted. 0 TNEAREF is set when the transmit FIFO has 16 bits or less 1 TNEAREF is set when the transmit FIFO has 32 bits or less
4 RNFULLF_ MARK	Receive FIFO nearly full watermark This bit selects the mark after which the RNFULLF flag is asserted. 0 RNFULLF is set when the receive FIFO has 48 bits or more 1 RNFULLF is set when the receive FIFO has 32 bits or more
3 INTCLR	Interrupt clearing mechanism select This bit selects the mechanism by which the SPRF, SPTEF, TNEAREF, and RNFULLF interrupts are cleared. 0 These interrupts are cleared when the corresponding flags are cleared depending on the state of the FIFOs 1 These interrupts are cleared by writing the corresponding bits in the CI register
2 TNEARIEN	Transmit FIFO nearly empty interrupt enable Writing 1 to this bit enables the SPI to interrupt the CPU when the TNEAREF flag is set. This bit is ignored and has no function if the FIFOMODE bit is 0. 0 No interrupt upon TNEAREF being set 1 Enable interrupts upon TNEAREF being set
1 RNFULLIEN	Receive FIFO nearly full interrupt enable Writing 1 to this bit enables the SPI to interrupt the CPU when the RNFULLF flag is set. This bit is ignored and has no function if the FIFOMODE bit is 0. 0 No interrupt upon RNFULLF being set 1 Enable interrupts upon RNFULLF being set
0 FIFOMODE	FIFO mode enable This bit enables the SPI to use a 64-bit FIFO (8 bytes or four 16-bit words) for both transmit and receive buffers. 0 Buffer mode disabled 1 Data available in the receive data buffer

31.4 Functional description

This section provides the functional description of the module.

31.4.1 General

The SPI system is enabled by setting the SPI enable (SPE) bit in SPI Control Register 1. While C1[SPE] is set, the four associated SPI port pins are dedicated to the SPI function as:

- Slave select (SS)
- Serial clock (SPSCK)
- Master out/slave in (MOSI)
- Master in/slave out (MISO)

An SPI transfer is initiated in the master SPI device by reading the SPI status register (SPIx_S) when S[SPTEF] = 1 and then writing data to the transmit data buffer (write to SPIx_DH:SPIx_DL). When a transfer is complete, received data is moved into the receive data buffer. The SPIx_DH:SPIx_DL registers act as the SPI receive data buffer for reads and as the SPI transmit data buffer for writes.

The Clock Phase Control (CPHA) and Clock Polarity Control (CPOL) bits in the SPI Control Register 1 (SPIx_C1) select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. C1[CPHA] is used to accommodate two fundamentally different protocols by sampling data on odd numbered SPSCK edges or on even numbered SPSCK edges.

The SPI can be configured to operate as a master or as a slave. When the MSTR bit in SPI Control Register 1 is set, master mode is selected; when C1[MSTR] is clear, slave mode is selected.

31.4.2 Master mode

The SPI operates in master mode when C1[MSTR] is set. Only a master SPI module can initiate transmissions. A transmission begins by reading the SPIx_S register while S[SPTEF] = 1 and writing to the master SPI data registers. If the shift register is empty, the byte immediately transfers to the shift register. The data begins shifting out on the MOSI pin under the control of the serial clock.

- SPSCK
  - The SPR3, SPR2, SPR1, and SPR0 baud rate selection bits in conjunction with the SPPR2, SPPR1, and SPPR0 baud rate preselection bits in the SPI Baud Rate register control the baud rate generator and determine the speed of the

transmission. The SPSCK pin is the SPI clock output. Through the SPSCK pin, the baud rate generator of the master controls the shift register of the slave peripheral.

- MOSI, MISO pin
  - In master mode, the function of the serial data output pin (MOSI) and the serial data input pin (MISO) is determined by the SPC0 and BIDIROE control bits.
- $\overline{\text{SS}}$ pin
  - If C2[MODFEN] and C1[SSOE] are set, the SS pin is configured as slave select output. The SS output becomes low during each transmission and is high when the SPI is in idle state. If C2[MODFEN] is set and C1[SSOE] is cleared, the $\overline{\text{SS}}$ pin is configured as input for detecting mode fault error. If the SS input becomes low this indicates a mode fault error where another master tries to drive the MOSI and SPSCK lines. In this case, the SPI immediately switches to slave mode by clearing C1[MSTR] and also disables the slave output buffer MISO (or SISO in bidirectional mode). As a result, all outputs are disabled, and SPSCK, MOSI and MISO are inputs. If a transmission is in progress when the mode fault occurs, the transmission is aborted and the SPI is forced into idle state. This mode fault error also sets the Mode Fault (MODF) flag in the SPI Status Register (SPIx_S). If the SPI Interrupt Enable bit (SPIE) is set when S[MODF] gets set, then an SPI interrupt sequence is also requested. When a write to the SPI Data Register in the master occurs, there is a half SPSCK-cycle delay. After the delay, SPSCK is started within the master. The rest of the transfer operation differs slightly, depending on the clock format specified by the SPI clock phase bit, CPHA, in SPI Control Register 1 (see SPI clock formats).

Note

A change of C1[CPOL], C1[CPHA], C1[SSOE], C1[LSBFE], C2[MODFEN], C2[SPC0], C2[BIDIROE] with C2[SPC0] set, SPIMODE, FIFOMODE, SPPR2-SPPR0 and SPR3-SPR0 in master mode abort a transmission in progress and force the SPI into idle state. The remote slave cannot detect this, therefore the master has to ensure that the remote slave is set back to idle state.

31.4.3 Slave mode

The SPI operates in slave mode when the MSTR bit in SPI Control Register 1 is clear.

- SPSCK

  In slave mode, SPSCK is the SPI clock input from the master.

- MISO, MOSI pin

  In slave mode, the function of the serial data output pin (MISO) and serial data input pin (MOSI) is determined by the SPC0 bit and BIDIROE bit in SPI Control Register 2.

- SS pin

  The SS pin is the slave select input. Before a data transmission occurs, the SS pin of the slave SPI must be low. SS must remain low until the transmission is complete. If SS goes high, the SPI is forced into an idle state.

  The SS input also controls the serial data output pin. If SS is high (not selected), the serial data output pin is high impedance. If SS is low, the first bit in the SPI Data Register is driven out of the serial data output pin. Also, if the slave is not selected (SS is high), then the SPSCK input is ignored and no internal shifting of the SPI shift register occurs.

  Although the SPI is capable of duplex operation, some SPI peripherals are capable of only receiving SPI data in a slave mode. For these simpler devices, there is no serial data out pin.

Note

When peripherals with duplex capability are used, take care not to simultaneously enable two receivers whose serial outputs drive the same system slave's serial data output line.

As long as no more than one slave device drives the system slave's serial data output line, it is possible for several slaves to receive the same transmission from a master, although the master would not receive return information from all of the receiving slaves.

If the CPHA bit in SPI Control Register 1 is clear, odd numbered edges on the SPSCK input cause the data at the serial data input pin to be latched. Even numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

If C1[CPHA] is set, even numbered edges on the SPSCK input cause the data at the serial data input pin to be latched. Odd numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on C1[LSBFE].

When C1[CPHA] is set, the first edge is used to get the first data bit onto the serial data output pin. When C1[CPHA] is clear and the SS input is low (slave selected), the first bit of the SPI data is driven out of the serial data output pin. After the eighth (SPIMODE = 0) or sixteenth (SPIMODE = 1) shift, the transfer is considered complete and the received data is transferred into the SPI Data register. To indicate transfer is complete, the SPRF flag in the SPI Status Register is set.

Note

A change of the bits FIFOMODE,SPIMODE, C2[BIDIROE] with C2[SPC0] set, C1[CPOL], C1[CPHA], C1[SSOE], C1[LSBFE], C2[MODFEN], and C2[SPC0] in slave mode will corrupt a transmission in progress and must be avoided.

31.4.4 SPI FIFO Mode

When the FIFO feature is supported: The SPI works in FIFO mode when the C3[FIFOMODE] bit is set. When the module is in FIFO mode, the SPI RX buffer and SPI TX buffer are replaced by an 8-byte-deep FIFO, as the following figures show.

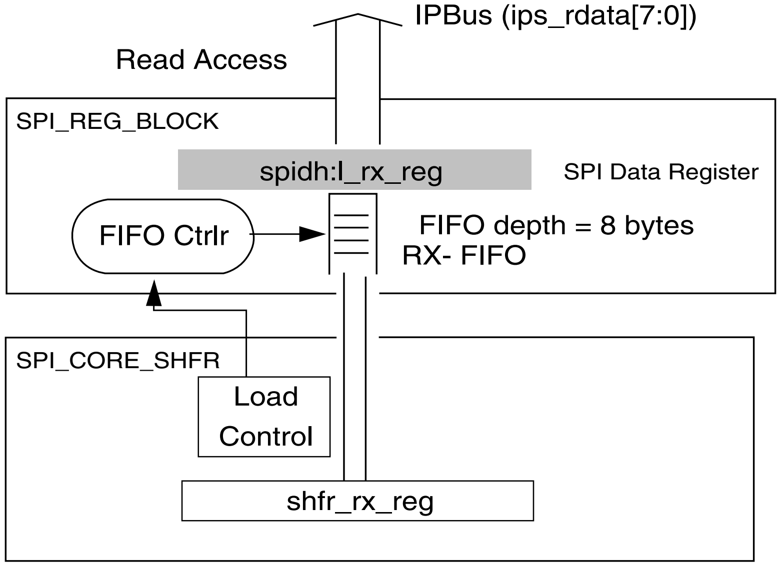


Figure 31-33. SPIH:L read side structural overview in FIFO mode

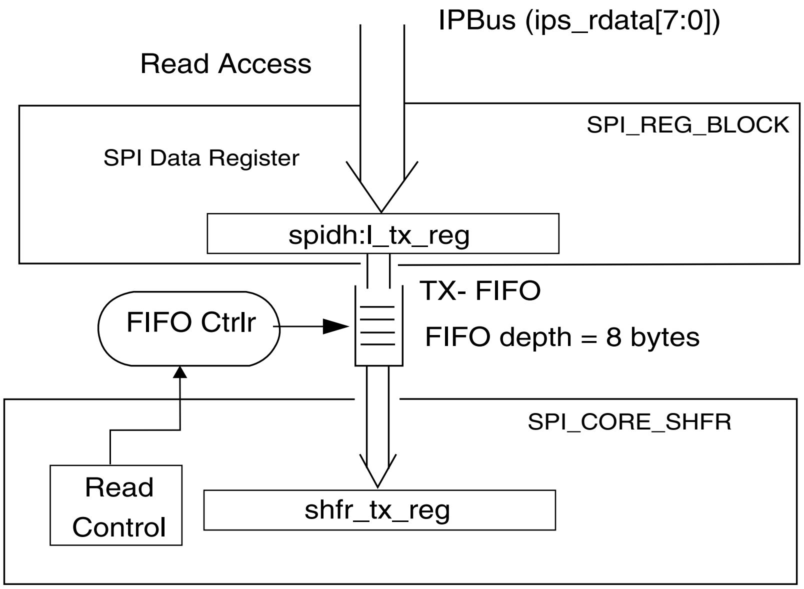


Figure 31-34. SPIH:L write side structural overview in FIFO mode

31.4.5 SPI Transmission by DMA

SPI supports both Transmit and Receive by DMA. The basic flow of SPI transmission by DMA is as below.

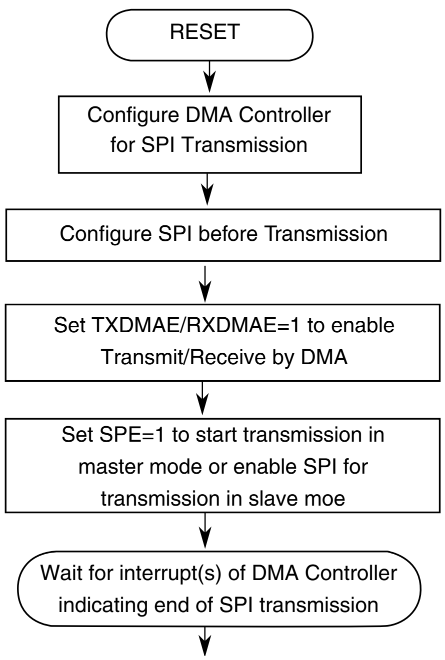


Figure 31-35. Basic Flow of SPI Transmission by DMA

31.4.5.1 Transmit by DMA

Transmit by DMA is supported only when TXDMAE is set. A transmit DMA request is asserted when both SPE and SPTEF are set. Then the on-chip DMA controller detects this request and transfers data from memory into the SPI data register. After that, TX DMA DONE is asserted to clear SPTEF automatically. This process repeats until all data for transmission (the number is decided by the configuration register[s] of the DMA controller) is sent.

When the FIFO feature is supported: In FIFO mode (FIFOMODE=1) and when a data length of 8 bits is selected (SPIMODE=0), the DMA transfer for one transmit DMA request can write more than 1 byte (up to 8 bytes) to the DL register because the TX FIFO can store 8 bytes of transmit data. In FIFO mode (FIFOMODE=1) and when a data length of 16 bits is selected (SPIMODE=1), the DMA transfer for one transmit DMA request can write more than 1 word (up to 4 words) to the DH:DL registers because the TX FIFO can store 4 words of transmit data. A larger number of bytes or words transferred from memory to the SPI data register for each transmit DMA request results in a lower total number of transmit DMA requests.

When FIFOMODE is 0: Cycle Steal (DMA_DCRn[CS] = 1) should be enabled when using the DMA controller to transfer data from memory to the SPI data register. The DMA performs a single data transfer per DMA request in cycle steal mode. Therefore, a single byte/word is written to the SPI data register from memory and transmitted by the SPI module for each DMA request, as long as the BCR value is greater than zero (DMA_DSR_BCRn[BCR] > 0). Once the BCR has reached zero, software must reconfigure the DMA controller if more data is to be transmitted. If a configuration error occurs (DMA_DSR_BCRn[CE] = 1) when the BCR is equal to 0, software must:

- disable peripheral requests when the BCR is equal to 0,
- perform 16-bit transfers (SPIMODE = 1), or
- decrease the SPI baud rate.

Software can disable peripheral requests by setting DMA_DCRn[D_REQ] = 1 when initializing the DMA controller, or by clearing DMA_DCRn[ERQ] once the BCR is equal to zero. Also, to continue transmitting data software must re-enable peripheral requests (DMA_DCRn[ERQ] = 1) after reconfiguring the DMA controller.

If continuous mode (DMA_DCRn[CS] = 0) is used when FIFOMODE = 0, the DMA controller transfers data continuously to the SPI data register once one DMA request is asserted; therefore, it is necessary to limit the BCR to 1 byte or 2 bytes for 8-bit and 16-bit modes, respectively. In addition, the initial SPI data transmission is repeated one or more times unless peripheral requests are disabled when the BCR is equal to zero or the first SPI transmit byte/word is written to the SPI data register prior to enabling DMA requests. Due to these limitations, continuous mode is not a practical configuration of the DMA controller to write data to the SPI data register and is not recommended.

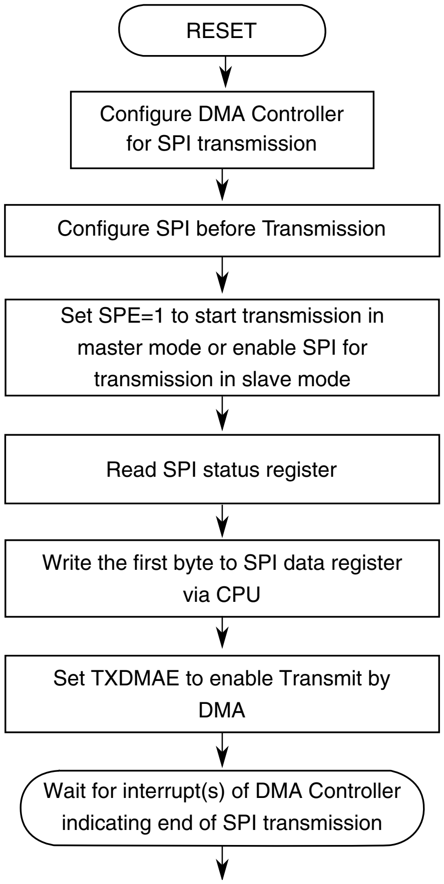


Figure 31-36. Recommended startup of SPI transmit by DMA

31.4.5.2 Receive by DMA

Receive by DMA is supported only when RXDMAE is set. A receive DMA request is asserted when both SPE and SPRF are set. Then the on-chip DMA controller detects this request and transfers data from the SPI data register into memory. After that, RX DMA DONE is asserted to clear SPRF automatically. This process repeats until all data to be received (the number is decided by configuration register[s] of the DMA controller) is received or no receive DMA request is generated again because the SPI transmission is finished.

When the FIFO feature is supported: In FIFO mode (FIFOMODE=1) and when a data length of 8 bits is selected (SPIMODE=0), the DMA transfer for one receive DMA request can read more than 1 byte (up to 8 bytes) from the SPI data register because the RX FIFO can hold 8 bytes. In FIFO mode (FIFOMODE=1) and when a data length of 16 bits is selected (SPIMODE=1), the DMA transfer for one receive DMA request can read more than 1 word (up to 4 words) from the DH:DL registers because the RX FIFO can hold 4 words. A larger number of bytes or words transferred from the SPI data register to memory for one receive DMA request results in a lower total number of receive DMA requests.

31.4.6 Data Transmission Length

The SPI can support data lengths of 8 or 16 bits. The length can be configured with the SPIMODE bit in the SPIx_C2 register.

In 8-bit mode (SPIMODE = 0), the SPI Data Register is comprised of one byte: SPIx_DL. The SPI Match Register is also comprised of only one byte: SPIx_ML. Reads of SPIx_DH and SPIx_MH will return zero. Writes to SPIx_DH and SPIx_MH will be ignored.

In 16-bit mode (SPIMODE = 1), the SPI Data Register is comprised of two bytes: SPIx_DH and SPIx_DL. Reading either byte (SPIx_DH or SPIx_DL) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. Writing to either byte (SPIx_DH or SPIx_DL) latches the value into a buffer. When both bytes have been written, they are transferred as a coherent 16-bit value into the transmit data buffer.

In 16-bit mode, the SPI Match Register is also comprised of two bytes: SPIx_MH and SPIx_ML. There is no buffer mechanism for the reading of SPIxMH and SPIxML since they can only be changed by writing at CPU side. Writing to either byte (SPIx_MH or SPIx_ML) latches the value into a buffer. When both bytes have been written, they are transferred as a coherent 16-bit value into the SPI Match Register.

Any switching between 8- and 16-bit data transmission length (controlled by SPIMODE bit) in master mode will abort a transmission in progress, force the SPI system into idle state, and reset all status bits in the SPIx_S register. To initiate a transfer after writing to SPIMODE, the SPIx_S register must be read with SPTEF = 1, and data must be written to SPIx_DH:SPIx_DL in 16-bit mode (SPIMODE = 1) or SPIx_DL in 8-bit mode (SPIMODE = 0).

In slave mode, user software should write to SPIMODE only once to prevent corrupting a transmission in progress.

Note

> Data can be lost if the data length is not the same for both master and slave devices.

31.4.7 SPI clock formats

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the SPI system has a Clock Polarity (CPOL) bit and a Clock Phase (CPHA) control bit in the Control Register 1 to select one of four clock formats for data transfers. C1[CPOL] selectively inserts an inverter in series with the clock. C1[CPHA] chooses between two different clock phase relationships between the clock and data.

The following figure shows the clock formats when SPIMODE = 0 (8-bit mode) and CPHA = 1. At the top of the figure, the eight bit times are shown for reference with bit 1 starting at the first SPSCK edge and bit 8 ending one-half SPSCK cycle after the eighth SPSCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting in LSBFE. Both variations of SPSCK polarity are shown, but only one of these waveforms applies for a specific transfer, depending on the value in C1[CPOL]. The SAMPLE IN waveform applies to the MOSI input of a slave or the MISO input of a master. The MOSI waveform applies to the MOSI output pin from a master and the MISO waveform applies to the MISO output from a slave. The $\overline{\text{SS}}$ OUT waveform applies to the slave select output from a master (provided C2[MODFEN] and C1[SSOE] = 1). The master $\overline{\text{SS}}$ output goes to active low one-half SPSCK cycle before the start of the transfer and goes back high at the end of the eighth bit time of the transfer. The $\overline{\text{SS}}$ IN waveform applies to the slave select input of a slave.

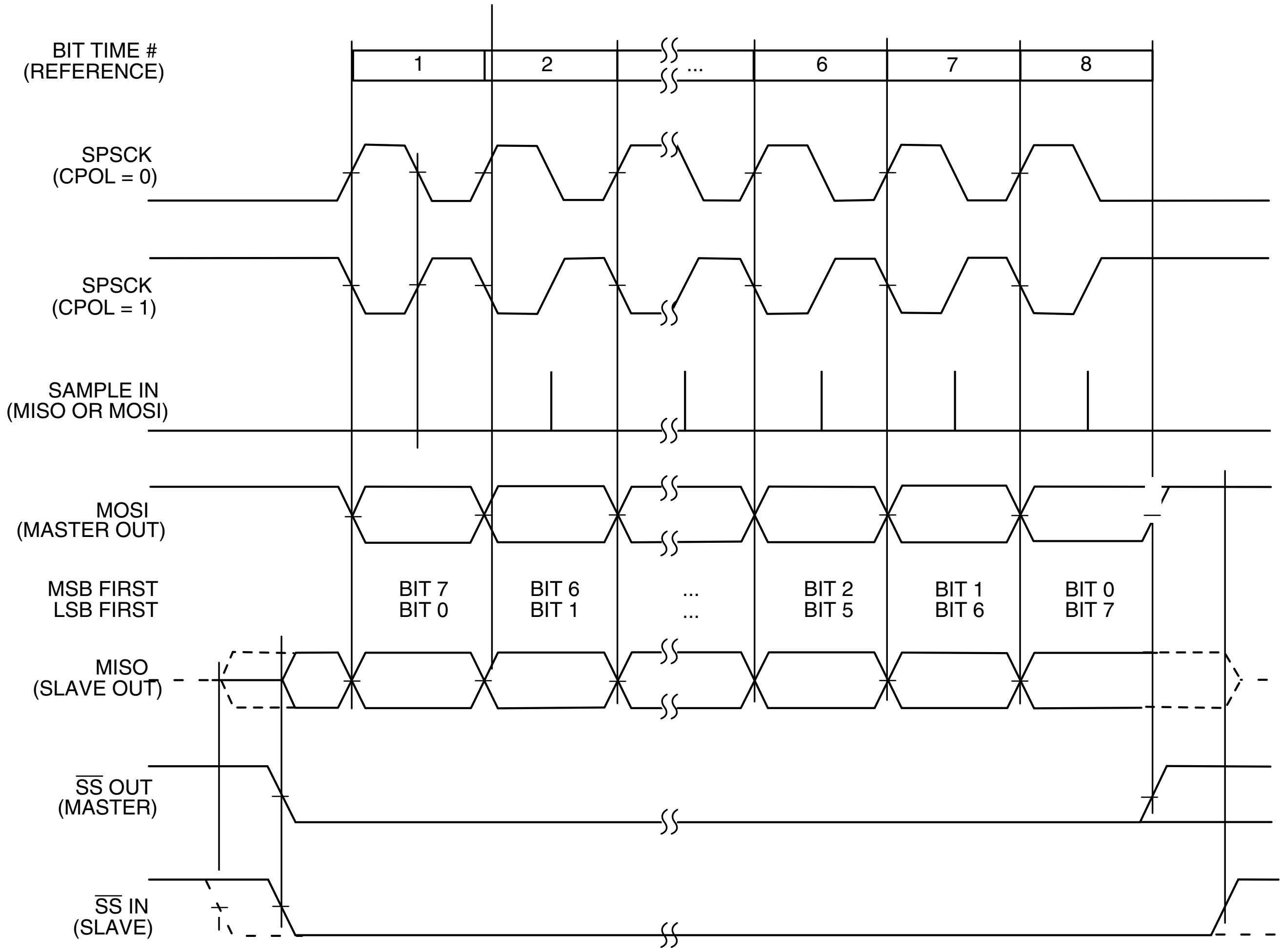


Figure 31-37. SPI clock formats (CPHA = 1)

When C1[CPHA] = 1, the slave begins to drive its MISO output when $\overline{SS}$ goes to active low, but the data is not defined until the first SPSCK edge. The first SPSCK edge shifts the first bit of data from the shifter onto the MOSI output of the master and the MISO output of the slave. The next SPSCK edge causes both the master and the slave to sample the data bit values on their MISO and MOSI inputs, respectively. At the third SPSCK edge, the SPI shifter shifts one bit position which shifts in the bit value that was just sampled, and shifts the second data bit value out the other end of the shifter to the MOSI and MISO outputs of the master and slave, respectively.

When C1[CPHA] = 1, the slave's $\overline{SS}$ input is not required to go to its inactive high level between transfers. In this clock format, a back-to-back transmission can occur, as follows:

1. A transmission is in progress.
2. A new data byte is written to the transmit buffer before the in-progress transmission is complete.
3. When the in-progress transmission is complete, the new, ready data byte is transmitted immediately.

Between these two successive transmissions, no pause is inserted; the $\overline{SS}$ pin remains low.

The following figure shows the clock formats when SPIMODE = 0 and C1[CPHA] = 0. At the top of the figure, the eight bit times are shown for reference with bit 1 starting as the slave is selected (SS IN goes low), and bit 8 ends at the last SPSCK edge. The MSB first and LSB first lines show the order of SPI data bits depending on the setting in LSBFE. Both variations of SPSCK polarity are shown, but only one of these waveforms applies for a specific transfer, depending on the value in CPOL. The SAMPLE IN waveform applies to the MOSI input of a slave or the MISO input of a master. The MOSI waveform applies to the MOSI output pin from a master and the MISO waveform applies to the MISO output from a slave. The SS OUT waveform applies to the slave select output from a master (provided C2[MODFEN] and C1[SSOE] = 1). The master SS output goes to active low at the start of the first bit time of the transfer and goes back high one-half SPSCK cycle after the end of the eighth bit time of the transfer. The SS IN waveform applies to the slave select input of a slave.

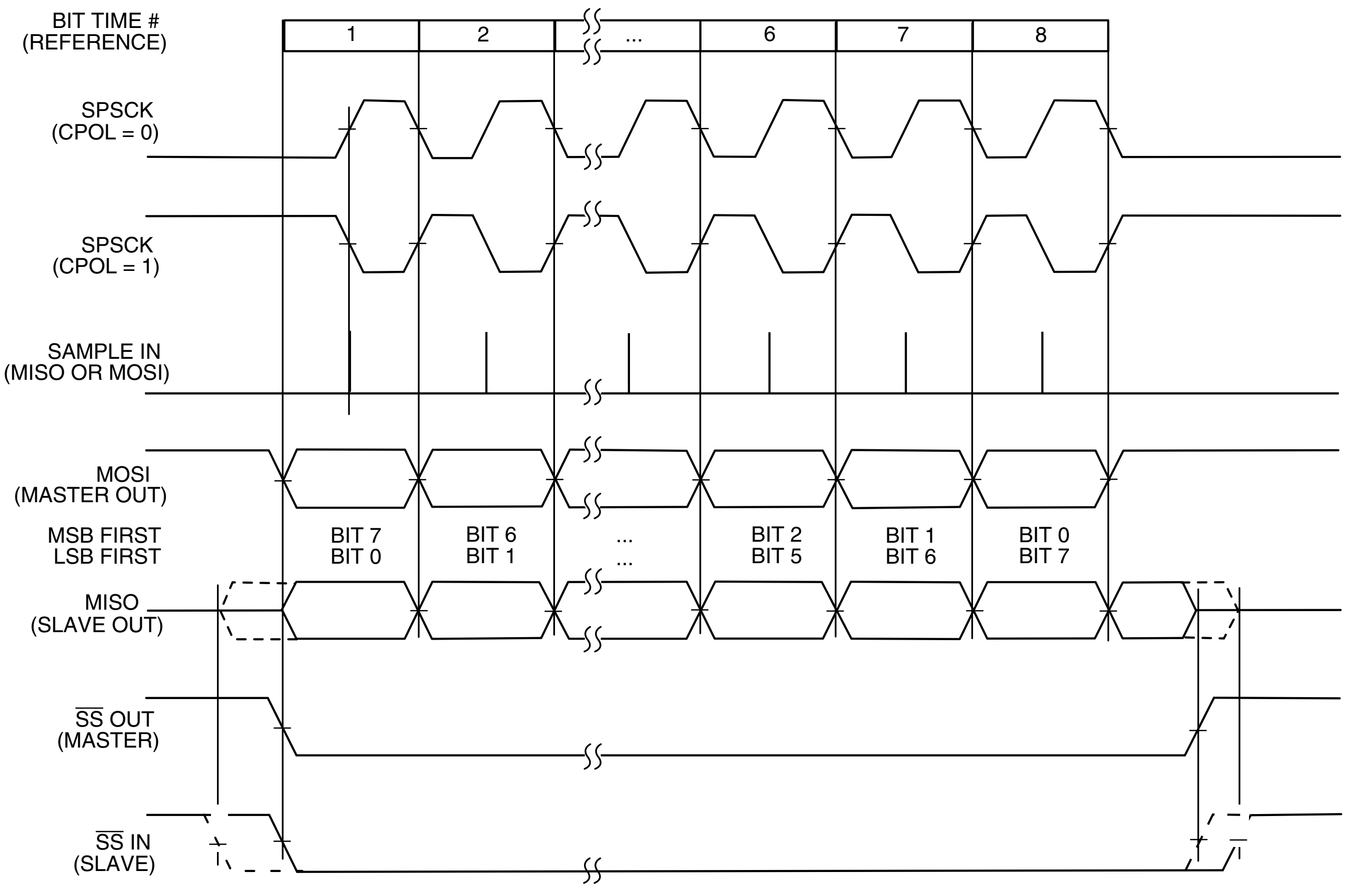


Figure 31-38. SPI clock formats (CPHA = 0)

When C1[CPHA] = 0, the slave begins to drive its MISO output with the first data bit value (MSB or LSB depending on LSBFE) when SS goes to active low. The first SPSCK edge causes both the master and the slave to sample the data bit values on their MISO and MOSI inputs, respectively. At the second SPSCK edge, the SPI shifter shifts one bit position which shifts in the bit value that was just sampled and shifts the second data bit value out the other end of the shifter to the MOSI and MISO outputs of the master and slave, respectively. When C1[CPHA] = 0, the slave's SS input must go to its inactive high level between transfers.

31.4.8 SPI baud rate generation

As shown in the following figure, the clock source for the SPI baud rate generator is the SPI module clock. The prescale bits (SPPR2:SPPR1:SPPR0) choose a prescale divisor of 1, 2, 3, 4, 5, 6, 7, or 8. The rate-select bits (SPR3:SPR2:SPR1:SPR0) divide the output of the prescaler stage by 2, 4, 8, 16, 32, 64, 128, 256, or 512 to get the internal SPI master mode bit-rate clock.

The baud rate generator is activated only when the SPI is in the master mode and a serial transfer is taking place. In the other cases, the divider is disabled to decrease I_{DD} current.

The baud rate divisor equation is as follows (except those reserved combinations in the SPI Baud Rate Divisor table).

```
BaudRateDivisor = (SPPR + 1) × 2^(SPR + 1)
```

The baud rate can be calculated with the following equation:

```
BaudRate = SPI Module Clock / BaudRateDivisor
```

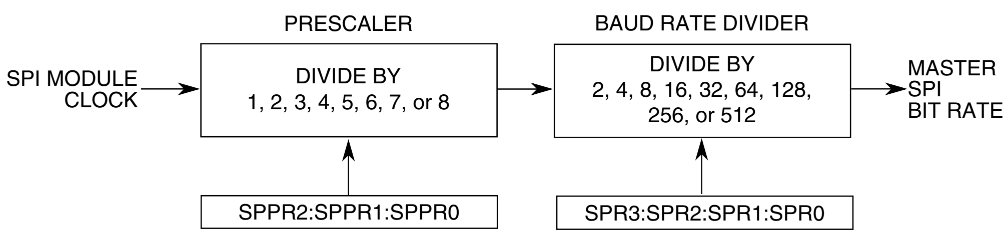


Figure 31-39. SPI baud rate generation

31.4.9 Special features

The following section describes the special features of SPI module.

31.4.9.1 $\overline{\text{SS}}$ Output

The $\overline{\text{SS}}$ output feature automatically drives the $\overline{\text{SS}}$ pin low during transmission to select external devices and drives the $\overline{\text{SS}}$ pin high during idle to deselect external devices. When the $\overline{\text{SS}}$ output is selected, the $\overline{\text{SS}}$ output pin is connected to the $\overline{\text{SS}}$ input pin of the external device.

The $\overline{\text{SS}}$ output is available only in master mode during normal SPI operation by asserting C1[SSOE] and C2[MODFEN] as shown in the description of C1[SSOE].

The mode fault feature is disabled while $\overline{\text{SS}}$ output is enabled.

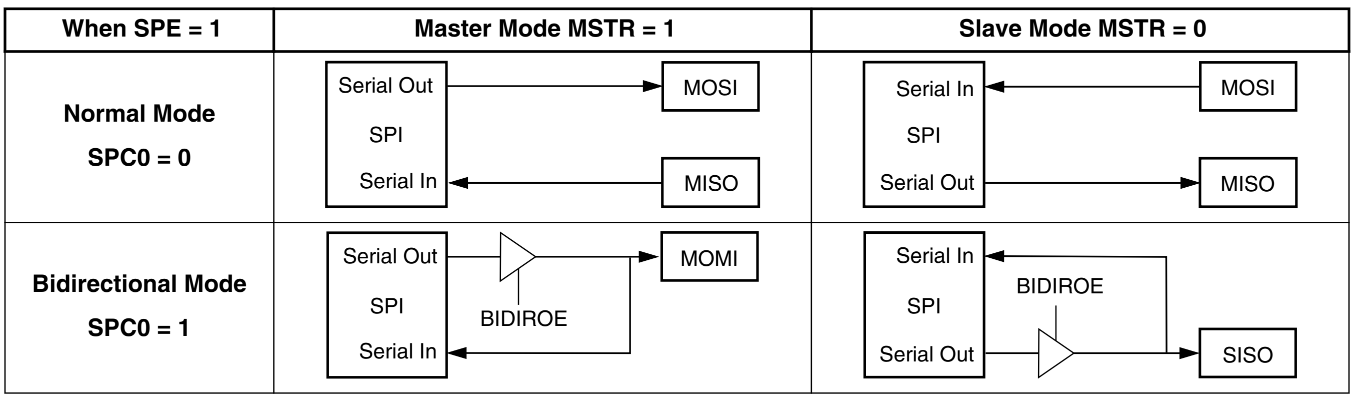
Note

> Be careful when using the $\overline{\text{SS}}$ output feature in a multimaster system because the mode fault feature is not available for detecting system errors between masters.

31.4.9.2 Bidirectional mode (MOMI or SISO)

The bidirectional mode is selected when the SPC0 bit is set in SPI Control Register 2 (see the following table). In this mode, the SPI uses only one serial data pin for the interface with one or more external devices. C1[MSTR] decides which pin to use. The MOSI pin becomes the serial data I/O (MOMI) pin for the master mode, and the MISO pin becomes serial data I/O (SISO) pin for the slave mode. The MISO pin in master mode and MOSI pin in slave mode are not used by the SPI.

Table 31-34. Normal Mode and Bidirectional Mode



The direction of each serial I/O pin depends on C2[BIDIROE]. If the pin is configured as an output, serial data from the shift register is driven out on the pin. The same pin is also the serial input to the shift register.

The SPSCK is an output for the master mode and an input for the slave mode.

$\overline{\text{SS}}$ is the input or output for the master mode, and it is always the input for the slave mode.

The bidirectional mode does not affect SPSCK and $\overline{\text{SS}}$ functions.

Note

In bidirectional master mode, with the mode fault feature enabled, both data pins MISO and MOSI can be occupied by the SPI, though MOSI is normally used for transmissions in bidirectional mode and MISO is not used by the SPI. If a mode fault occurs, the SPI is automatically switched to slave mode. In this case, MISO becomes occupied by the SPI and MOSI is not used. Consider this scenario if the MISO pin is used for another purpose.

31.4.10 Error conditions

The SPI module has one error condition: the mode fault error.

31.4.10.1 Mode fault error

If the $\overline{\text{SS}}$ input becomes low while the SPI is configured as a master, it indicates a system error where more than one master may be trying to drive the MOSI and SPSCK lines simultaneously. This condition is not permitted in normal operation, and it sets the MODF bit in the SPI status register automatically provided that C2[MODFEN] is set.

In the special case where the SPI is in master mode and C2[MODFEN] is cleared, the $\overline{\text{SS}}$ pin is not used by the SPI. In this special case, the mode fault error function is inhibited and MODF remains cleared. If the SPI system is configured as a slave, the $\overline{\text{SS}}$ pin is a dedicated input pin. A mode fault error does not occur in slave mode.

If a mode fault error occurs, the SPI is switched to slave mode, with the exception that the slave output buffer is disabled. So the SPSCK, MISO and MOSI pins are forced to be high impedance inputs to avoid any possibility of conflict with another output driver. A transmission in progress is aborted and the SPI is forced into idle state.

If the mode fault error occurs in the bidirectional mode for an SPI system configured in master mode, the output enable of MOMI (MOSI in bidirectional mode) is cleared if it was set. No mode fault error occurs in the bidirectional mode for the SPI system configured in slave mode.

The mode fault flag is cleared automatically by a read of the SPI Status Register (with MODF set) followed by a write to SPI Control Register 1. If the mode fault flag is cleared, the SPI becomes a normal master or slave again.

31.4.11 Low-power mode options

This section describes the low-power mode options.

31.4.11.1 SPI in Run mode

In Run mode, with the SPI system enable (SPE) bit in the SPI Control Register 1 clear, the SPI system is in a low-power, disabled state. SPI registers can still be accessed, but clocks to the core of this module are disabled.

31.4.11.2 SPI in Wait mode

SPI operation in Wait mode depends upon the state of the SPISWAI bit in SPI Control Register 2.

- If C2[SPISWAI] is clear, the SPI operates normally when the CPU is in Wait mode.
- If C2[SPISWAI] is set, SPI clock generation ceases and the SPI module enters a power conservation state when the CPU is in wait mode.
  - If C2[SPISWAI] is set and the SPI is configured for master, any transmission and reception in progress stops at Wait mode entry. The transmission and reception resumes when the SPI exits Wait mode.
  - If C2[SPISWAI] is set and the SPI is configured as a slave, any transmission and reception in progress continues if the SPSCK continues to be driven from the master. This keeps the slave synchronized to the master and the SPSCK.

    If the master transmits data while the slave is in wait mode, the slave continues to send data consistent with the operation mode at the start of wait mode (that is, if the slave is currently sending its SPIx_DH:SPIx_DL to the master, it continues to send the same byte. Otherwise, if the slave is currently sending the last data received byte from the master, it continues to send each previously received data from the master byte).

Note

Care must be taken when expecting data from a master while the slave is in a Wait mode or a Stop mode where the peripheral bus clock is stopped but internal logic states are retained. Even though the shift register continues to operate, the rest of the SPI is shut down (that is, an SPRF interrupt is not generated until an exit from Stop or Wait mode). Also, the data from the shift register is not copied into the SPIx_DH:SPIx_DL registers until after the slave SPI has exited Wait or Stop mode. An SPRF flag and SPIx_DH:SPIx_DL copy is only generated if Wait mode is entered or exited during a transmission. If the slave enters Wait mode in idle mode and exits Wait mode in idle mode, neither an SPRF nor a SPIx_DH:SPIx_DL copy occurs.

31.4.11.3 SPI in Stop mode

Operation in a Stop mode where the peripheral bus clock is stopped but internal logic states are retained depends on the SPI system. The Stop mode does not depend on C2[SPISWAI]. Upon entry to this type of stop mode, the SPI module clock is disabled (held high or low).

- If the SPI is in master mode and exchanging data when the CPU enters the Stop mode, the transmission is frozen until the CPU exits stop mode. After the exit from stop mode, data to and from the external SPI is exchanged correctly.
- In slave mode, the SPI remains synchronized with the master.

The SPI is completely disabled in a stop mode where the peripheral bus clock is stopped and internal logic states are not retained. After an exit from this type of stop mode, all registers are reset to their default values, and the SPI module must be reinitialized.

31.4.12 Reset

The reset values of registers and signals are described in the Memory Map and Register Descriptions content, which details the registers and their bitfields.

- If a data transmission occurs in slave mode after a reset without a write to SPIx_DH:SPIx_DL, the transmission consists of "garbage" or the data last received from the master before the reset.
- Reading from SPIx_DH:SPIx_DL after reset always returns zeros.

31.4.13 Interrupts

The SPI originates interrupt requests only when the SPI is enabled (the SPE bit in the SPIx_C1 register is set). The following is a description of how the SPI makes a request and how the MCU should acknowledge that request. The interrupt vector offset and interrupt priority are chip dependent.

Four flag bits, three interrupt mask bits, and one interrupt vector are associated with the SPI system. The SPI interrupt enable mask (SPIE) enables interrupts from the SPI receiver full flag (SPRF) and mode fault flag (MODF). The SPI transmit interrupt enable mask (SPTIE) enables interrupts from the SPI transmit buffer empty flag (SPTEF). The SPI match interrupt enable mask bit (SPIMIE) enables interrupts from the SPI match flag (SPMF). When one of the flag bits is set, and the associated interrupt mask bit is set, a hardware interrupt request is sent to the CPU. If the interrupt mask bits are cleared, software can poll the associated flag bits instead of using interrupts. The SPI interrupt service routine (ISR) should check the flag bits to determine which event caused the interrupt. The service routine should also clear the flag bit(s) before returning from the ISR (usually near the beginning of the ISR).

31.4.13.1 MODF

MODF occurs when the master detects an error on the $\overline{SS}$ pin. The master SPI must be configured for the MODF feature (see the description of the C1[SSOE] bit). Once MODF is set, the current transfer is aborted and the master (MSTR) bit in the SPIx_C1 register resets to 0.

The MODF interrupt is reflected in the status register's MODF flag. Clearing the flag also clears the interrupt. This interrupt stays active while the MODF flag is set. MODF has an automatic clearing process that is described in the SPI Status Register.

31.4.13.2 SPRF

SPRF occurs when new data has been received and copied to the SPI receive data buffer. In 8-bit mode, SPRF is set only after all 8 bits have been shifted out of the shift register and into SPIx_DL. In 16-bit mode, SPRF is set only after all 16 bits have been shifted out of the shift register and into SPIx_DH:SPIx_DL.

After SPRF is set, it does not clear until it is serviced. SPRF has an automatic clearing process that is described in the SPI Status Register details. If the SPRF is not serviced before the end of the next transfer (that is, SPRF remains active throughout another transfer), the subsequent transfers are ignored and no new data is copied into the Data register.

31.4.13.3 SPTEF

SPTEF occurs when the SPI transmit buffer is ready to accept new data. In 8-bit mode, SPTEF is set only after all 8 bits have been moved from SPIx_DL into the shifter. In 16-bit mode, SPTEF is set only after all 16 bits have been moved from SPIx_DH:SPIx_DL into the shifter.

After SPTEF is set, it does not clear until it is serviced. SPTEF has an automatic clearing process that is described in the SPI Status Register details.

31.4.13.4 SPMF

SPMF occurs when the data in the receive data buffer is equal to the data in the SPI Match Register. In 8-bit mode, SPMF is set only after bits 7–0 in the receive data buffer are determined to be equivalent to the value in SPIx_ML. In 16-bit mode, SPMF is set after bits 15–0 in the receive data buffer are determined to be equivalent to the value in SPIx_MH:SPIx_ML.

31.4.13.5 TNEAREF

The TNEAREF bit applies when the FIFO feature is supported.

The TNEAREF flag is set when only one 16-bit word or two 8-bit bytes of data remain in the transmit FIFO provided C3[5] = 0 or when only two 16-bit words or four 8-bit bytes of data remain in the transmit FIFO provided C3[5] =1. If FIFOMODE is not enabled, ignore this bit.

Clearing this interrupt depends on the state of C3[3] and the status of TNEAREF. Refer to the description of the SPI status (S) register.

31.4.13.6 RNFULLF

The RNFULLF bit applies when the FIFO feature is supported.

RNFULLF is set when more than three 16-bit words or six 8-bit bytes of data remain in the receive FIFO provided C3[4] = 0 or when more than two 16-bit words or four 8-bit bytes of data remain in the receive FIFO provided C3[4] = 1.

Clearing this interrupt depends on the state of C3[3] and the status of RNFULLF. Refer to the description of the SPI status (S) register.

31.4.13.7 Asynchronous interrupt in low-power modes

When the CPU is in Wait mode or Stop mode and the SPI module receives a transmission, the SPI module can generate an asynchronous interrupt to wake the CPU from the low power mode. The module generates the asynchronous interrupt only when all of the following conditions apply:

1. C2[SPLPIE] is set to 1.
2. The CPU is in Wait mode—in which case C2[SPISWAI] must be 1—or in Stop mode where the peripheral bus clock is stopped but internal logic states are retained.
3. The SPI module is in slave mode.
4. The received transmission ends.
5. When the FIFO feature is supported, FIFO mode is disabled: C3[FIFOMODE] is 0.

After the interrupt wakes the CPU and the peripheral bus clock is active again, the SPI module copies the received data from the shifter into the Data register and generates flags or DMA request signals. During the wakeup phase, a continuous transmission from a master would destroy the first received data.

31.5 Initialization/application information

This section discusses an example of how to initialize and use the SPI.

NOTE

When operating the SPI at the maximum baud rate it must be configured for 16 bit operation.

31.5.1 Initialization sequence

Before the SPI module can be used for communication, an initialization procedure must be carried out, as follows:

1. Update the Control Register 1 (SPIx_C1) to enable the SPI and to control interrupt enables. This register also sets the SPI as master or slave, determines clock phase and polarity, and configures the main SPI options.
2. Update the Control Register 2 (SPIx_C2) to enable additional SPI functions such as the SPI match interrupt feature, the master mode-fault function, and bidirectional mode output as well as to control 8- or 16-bit mode selection and other optional features.
3. Update the Baud Rate Register (SPIx_BR) to set the prescaler and bit rate divisor for an SPI master.
4. Update the Hardware Match Register (SPIx_MH:SPIx_ML) with the value to be compared to the receive data register for triggering an interrupt if hardware match interrupts are enabled.
5. In the master, read SPIx_S while S[SPTEF] = 1, and then write to the transmit data register (SPIx_DH:SPIx_DL) to begin transfer.

31.5.2 Pseudo-Code Example

In this example, the SPI module is set up for master mode with only hardware match interrupts enabled. The SPI runs in 16-bit mode at a maximum baud rate of SPI module clock divided by 2. Clock phase and polarity are set for an active-high SPI clock where the first edge on SPSCK occurs at the start of the first cycle of a data transfer.

SPIx_C1=0x54(%01010100)					
	Bit 7	SPIE	=	0	Disables receive and mode fault interrupts
	Bit 6	SPE	=	1	Enables the SPI system
	Bit 5	SPTIE	=	0	Disables SPI transmit interrupts
	Bit 4	MSTR	=	1	Sets the SPI module as a master SPI device
	Bit 3	CPOL	=	0	Configures SPI clock as active-high
	Bit 2	CPHA	=	1	First edge on SPSCK at start of first data transfer cycle
	Bit 1	SSOE	=	0	Determines SS pin function when mode fault enabled
	Bit 0	LSBFE	=	0	SPI serial data transfers start with most significant bit

SPIx_C2 = 0xC0(%11000000)					
	Bit 7	SPMIE	=	1	SPI hardware match interrupt enabled
	Bit 6	SPIMODE	=	1	Configures SPI for 16-bit mode
	Bit 5	TXDMAE	=	0	DMA request disabled
	Bit 4	MODFEN	=	0	Disables mode fault function

Table continues on the next page...

SPIx_C2 = 0xC0(%11000000)					
	Bit 3	BIDIROE	=	0	SPI data I/O pin acts as input
	Bit 2	RXDMAE	=	0	DMA request disabled
	Bit 1	SPISWAI	=	0	SPI clocks operate in wait mode
	Bit 0	SPC0	=	0	uses separate pins for data input and output

SPIx_BR = 0x00(%00000000)					
	Bit 7		=	0	Reserved
	Bit 6:4		=	000	Sets prescale divisor to 1
	Bit 3:0		=	0000	Sets baud rate divisor to 2

SPIx_S = 0x00(%00000000)					
	Bit 7	SPRF	=	0	Flag is set when receive data buffer is full
	Bit 6	SPMF	=	0	Flag is set when SPIx_MH/ML = receive data buffer
	Bit 5	SPTEF	=	0	Flag is set when transmit data buffer is empty
	Bit 4	MODF	=	0	Mode fault flag for master mode
	Bit 3:0		=	0	FIFOMODE is not enabled

SPIx_MH = 0xXX	
	In 16-bit mode, this register holds bits 8–15 of the hardware match buffer. In 8-bit mode, writes to this register will be ignored.

SPIx_ML = 0xXX	
	Holds bits 0–7 of the hardware match buffer.

SPIx_DH = 0xxx	
	In 16-bit mode, this register holds bits 8–15 of the data to be transmitted by the transmit buffer and received by the receive buffer.

SPIx_DL = 0xxx	
	Holds bits 0–7 of the data to be transmitted by the transmit buffer and received by the receive buffer.

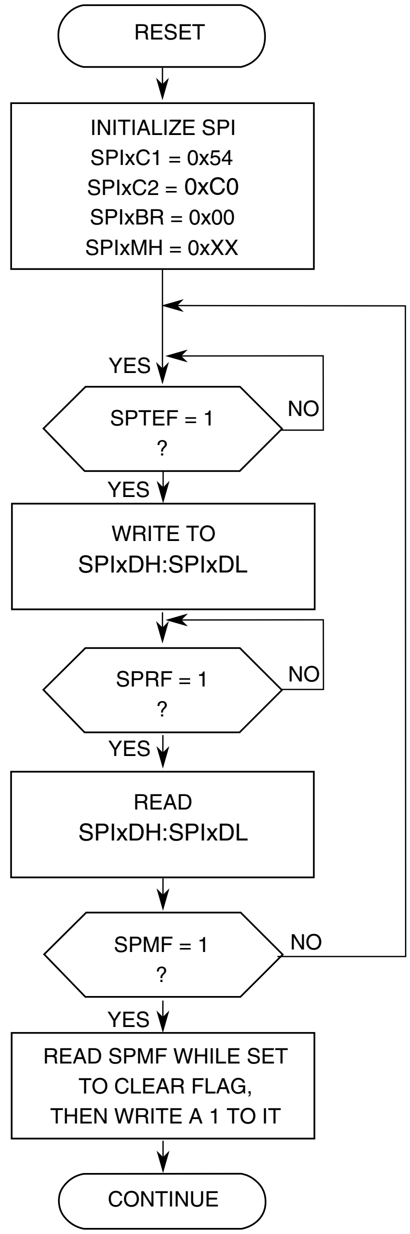


Figure 31-40. Initialization Flowchart Example for SPI Master Device in 16-bit Mode for FIFOMODE = 0

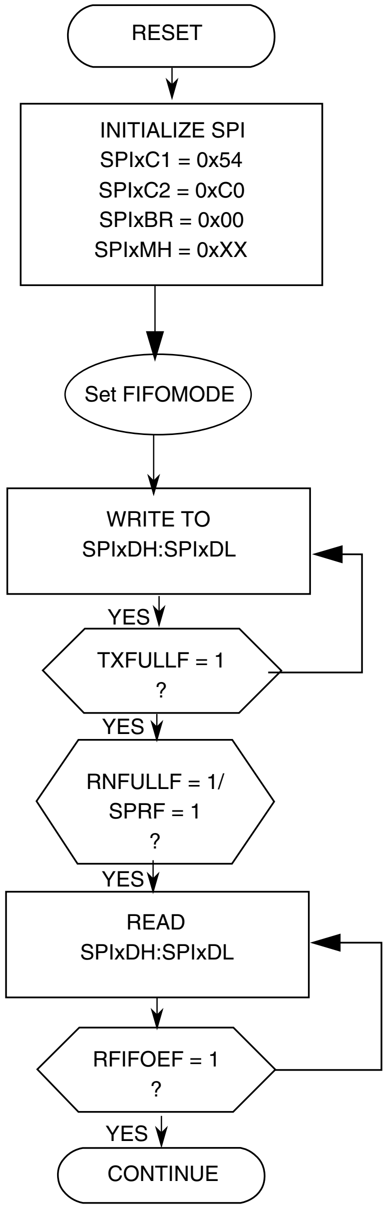


Figure 31-41. Initialization Flowchart Example for SPI Master Device in 16-bit Mode for FIFOMODE = 1

Chapter 32
Universal asynchronous receiver/transmitter (UART)

32.1 Introduction

32.1.1 Features

Features of the UART module include:

- Full-duplex, standard non-return-to-zero (NRZ) format
- Double-buffered transmitter and receiver with separate enables
- Programmable baud rates (13-bit modulo divider)
- 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
- Configurable receiver buad rate oversampling ratio from 4x to 32x
- 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
- Hardware parity generation and checking
- Programmable 8-bit, 9-bit or 10-bit character length
- Programmable 1-bit or 2-bit stop bits
- Receiver wakeup by idle-line, address-mark or address match
- Optional 13-bit break character generation / 11-bit break character detection
- Selectable transmitter output and receiver input polarity

32.1.2 Modes of operation

32.1.2.1 Stop mode

The UART will remain functional during Stop mode, provided the asynchronous transmit and receive clock remains enabled. The UART can generate an interrupt or DMA request to cause a wakeup from Stop mode.

32.1.2.2 Wait mode

The UART 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.

32.1.2.3 Debug mode

The UART remains functional in debug mode.

32.1.3 Block diagram

The following figure shows the transmitter portion of the UART.

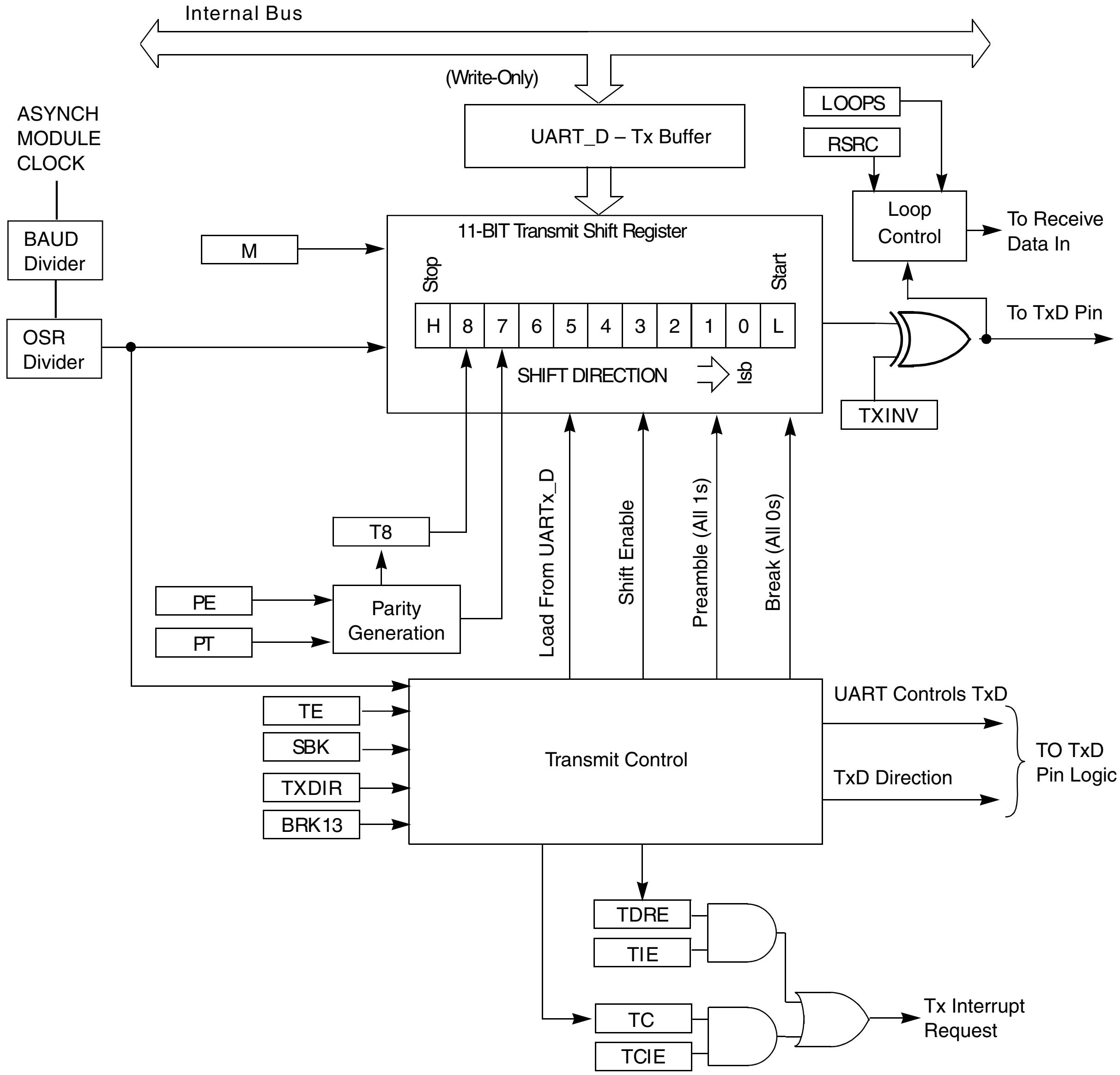


Figure 32-1. UART transmitter block diagram

The following figure shows the receiver portion of the UART.

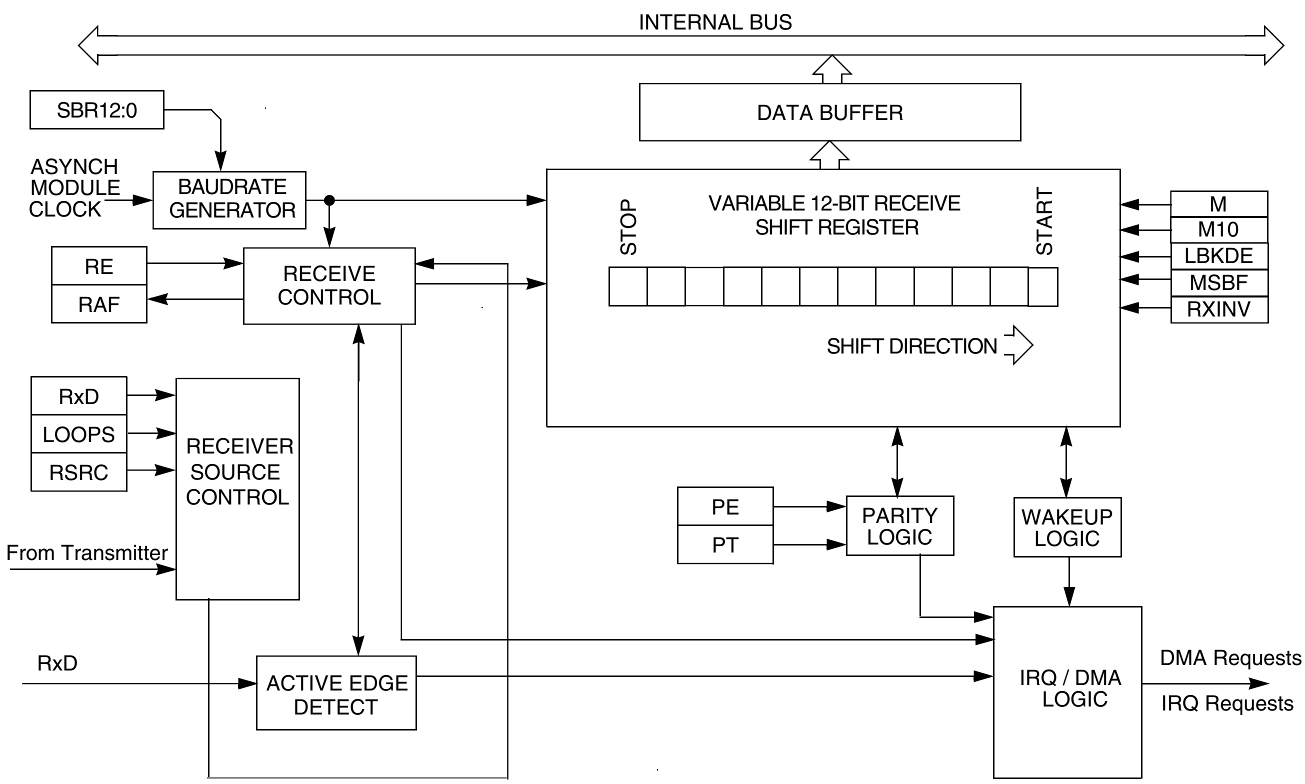


Figure 32-2. UART receiver block diagram

32.2 Register definition

The UART includes registers to control baud rate, select UART options, report UART status, and for transmit/receive data.

Accesses to address outside the valid memory map will generate a bus error.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_A000	UART Baud Rate Register High (UART0_BDH)	8	R/W	00h	32.2.1/613
4006_A001	UART Baud Rate Register Low (UART0_BDL)	8	R/W	04h	32.2.2/614
4006_A002	UART Control Register 1 (UART0_C1)	8	R/W	00h	32.2.3/614
4006_A003	UART Control Register 2 (UART0_C2)	8	R/W	00h	32.2.4/616
4006_A004	UART Status Register 1 (UART0_S1)	8	R/W	C0h	32.2.5/617
4006_A005	UART Status Register 2 (UART0_S2)	8	R/W	00h	32.2.6/619
4006_A006	UART Control Register 3 (UART0_C3)	8	R/W	00h	32.2.7/621
4006_A007	UART Data Register (UART0_D)	8	R/W	00h	32.2.8/622

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_A008	UART Match Address Registers 1 (UART0_MA1)	8	R/W	00h	32.2.9/623
4006_A009	UART Match Address Registers 2 (UART0_MA2)	8	R/W	00h	32.2.10/ 624
4006_A00A	UART Control Register 4 (UART0_C4)	8	R/W	0Fh	32.2.11/ 624
4006_A00B	UART Control Register 5 (UART0_C5)	8	R/W	00h	32.2.12/ 625

32.2.1 UART Baud Rate Register High (UARTx_BDH)

This register, along with UART_BDL, controls the prescale divisor for UART baud rate generation. The 13-bit baud rate setting [SBR12:SBR0] should only be updated when the transmitter and receiver are both disabled.

Address: Base address + 0h offset

Bit	7	6	5	4	3	2	1	0
Read / Write	LBKDIE	RXEDGIE	SBNS	SBR				
Reset	0	0	0	0	0	0	0	0

UARTx_BDH field descriptions

Field	Description
7 LBKDIE	LIN Break Detect Interrupt Enable (for LBKDIF) 0 Hardware interrupts from UART_S2[LBKDIF] disabled (use polling). 1 Hardware interrupt requested when UART_S2[LBKDIF] flag is 1.
6 RXEDGIE	RX Input Active Edge Interrupt Enable (for RXEDGIF) 0 Hardware interrupts from UART_S2[RXEDGIF] disabled (use polling). 1 Hardware interrupt requested when UART_S2[RXEDGIF] flag is 1.
5 SBNS	Stop Bit Number Select SBNS determines whether data characters are one or two stop bits. This bit should only be changed when the transmitter and receiver are both disabled. 0 One stop bit. 1 Two stop bit.
4–0 SBR	Baud Rate Modulo Divisor. The 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the baud rate generator. When BR is 1 - 8191, the baud rate equals baud clock / ((OSR+1) × BR).

32.2.2 UART Baud Rate Register Low (UARTx_BDL)

This register, along with UART_BDH, control the prescale divisor for UART baud rate generation. The 13-bit baud rate setting [SBR12:SBR0] can only be updated when the transmitter and receiver are both disabled.

UART_BDL is reset to a non-zero value, so after reset the baud rate generator remains disabled until the first time the receiver or transmitter is enabled; that is, UART_C2[RE] or UART_C2[TE] bits are written to 1.

Address: Base address + 1h offset

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

UARTx_BDL field descriptions

Field	Description
7–0 SBR	Baud Rate Modulo Divisor These 13 bits in SBR[12:0] are referred to collectively as BR. They set the modulo divide rate for the baud rate generator. When BR is 1 - 8191, the baud rate equals baud clock/((OSR+1) × BR).

32.2.3 UART Control Register 1 (UARTx_C1)

This read/write register controls various optional features of the UART system. This register should only be altered when the transmitter and receiver are both disabled.

Address: Base address + 2h offset

Bit	7	6	5	4	3	2	1	0
Read Write	LOOPS	DOZEEN	RSRC	M	WAKE	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0

UARTx_C1 field descriptions

Field	Description
7 LOOPS	Loop Mode Select Selects between loop back modes and normal 2-pin full-duplex modes. When LOOPS is set, the transmitter output is internally connected to the receiver input.

Table continues on the next page...

UARTx_C1 field descriptions (continued)

Field	Description
	0 Normal operation - UART_RX and UART_TX use separate pins. 1 Loop mode or single-wire mode where transmitter outputs are internally connected to receiver input. (See RSRC bit.) UART_RX pin is not used by UART.
6 DOZEEN	Doze Enable 0 UART is enabled in Wait mode. 1 UART is disabled in Wait mode.
5 RSRC	Receiver Source Select This bit has no meaning or effect unless the LOOPS bit is set to 1. When LOOPS is set, the receiver input is internally connected to the UART_TX pin and RSRC determines whether this connection is also connected to the transmitter output. 0 Provided LOOPS is set, RSRC is cleared, selects internal loop back mode and the UART does not use the UART_RX pins. 1 Single-wire UART mode where the UART_TX pin is connected to the transmitter output and receiver input.
4 M	9-Bit or 8-Bit Mode Select 0 Receiver and transmitter use 8-bit data characters. 1 Receiver and transmitter use 9-bit data characters.
3 WAKE	Receiver Wakeup Method Select 0 Idle-line wakeup. 1 Address-mark wakeup.
2 ILT	Idle Line Type Select Setting this bit to 1 ensures that the stop bits and logic 1 bits at the end of a character do not count toward the 10 to 13 bit times of logic high level needed by the idle line detection logic. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.
1 PE	Parity Enable Enables hardware parity generation and checking. When parity is enabled, the bit immediately before the stop bit is treated as the parity bit. 0 No hardware parity generation or checking. 1 Parity enabled.
0 PT	Parity Type Provided parity is enabled (PE = 1), this bit selects even or odd parity. Odd parity means the total number of 1s in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even. 0 Even parity. 1 Odd parity.

32.2.4 UART Control Register 2 (UARTx_C2)

This register can be read or written at any time.

Address: Base address + 3h offset

Bit	7	6	5	4	3	2	1	0
Read / Write	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0

UARTx_C2 field descriptions

Field	Description
7 TIE	Transmit Interrupt Enable for TDRE 0 Hardware interrupts from TDRE disabled; use polling. 1 Hardware interrupt requested when TDRE flag is 1.
6 TCIE	Transmission Complete Interrupt Enable for TC 0 Hardware interrupts from TC disabled; use polling. 1 Hardware interrupt requested when TC flag is 1.
5 RIE	Receiver Interrupt Enable for RDRF 0 Hardware interrupts from RDRF disabled; use polling. 1 Hardware interrupt requested when RDRF flag is 1.
4 ILIE	Idle Line Interrupt Enable for IDLE 0 Hardware interrupts from IDLE disabled; use polling. 1 Hardware interrupt requested when IDLE flag is 1.
3 TE	Transmitter Enable TE must be 1 to use the UART transmitter. When TE is set, the UART forces the UART_TX pin to act as an output for the UART system. When the UART is configured for single-wire operation (LOOPS = RSRC = 1), TXDIR controls the direction of traffic on the single UART communication line (UART_TX pin). TE can also queue an idle character by clearing TE then setting TE while a transmission is in progress. When TE is written to 0, the transmitter keeps control of the port UART_TX pin until any data, queued idle, or queued break character finishes transmitting before allowing the pin to tristate. 0 Transmitter disabled. 1 Transmitter enabled.
2 RE	Receiver Enable When the UART receiver is off or LOOPS is set, the UART_RX pin is not used by the UART . When RE is written to 0, the receiver finishes receiving the current character (if any). 0 Receiver disabled. 1 Receiver enabled.
1 RWU	Receiver Wakeup Control

Table continues on the next page...

UARTx_C2 field descriptions (continued)

Field	Description
	This bit can be written to 1 to place the UART receiver in a standby state where it waits for automatic hardware detection of a selected wakeup condition. The wakeup condition is an idle line between messages, WAKE = 0, idle-line wakeup, or a logic 1 in the most significant data bit in a character, WAKE = 1, address-mark wakeup. Application software sets RWU and, normally, a selected hardware condition automatically clears RWU. 0 Normal UART receiver operation. 1 UART receiver in standby waiting for wakeup condition.
0 SBK	Send Break Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 to 13, or 13 to 16 if BRK13 = 1, bit times of logic 0 are queued as long as SBK is set. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. 0 Normal transmitter operation. 1 Queue break character(s) to be sent.

32.2.5 UART Status Register 1 (UARTx_S1)

Address: Base address + 4h offset

Bit	7	6	5	4	3	2	1	0
Read	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
Write				w1c	w1c	w1c	w1c	w1c
Reset	1	1	0	0	0	0	0	0

UARTx_S1 field descriptions

Field	Description
7 TDRE	Transmit Data Register Empty Flag TDRE is set out of reset and whenever there is room to write data to the transmit data buffer. To clear TDRE, write to the UART data register (UART_D). 0 Transmit data buffer full. 1 Transmit data buffer empty.
6 TC	Transmission Complete Flag TC is set out of reset and when TDRE is set and no data, preamble, or break character is being transmitted. TC is cleared automatically by one of the following: • Write to the UART data register (UART_D) to transmit new data • Queue a preamble by changing TE from 0 to 1 • Queue a break character by writing 1 to UART_C2[SBK] 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete).

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
5 RDRF	Receive Data Register Full Flag RDRF becomes set whenever the receive data buffer is full. To clear RDRF, read the UART data register (UART_D). 0 Receive data buffer empty. 1 Receive data buffer full.
4 IDLE	Idle Line Flag IDLE is set when the UART 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 received and RDRF has been set. IDLE is set only once even if the receive line remains idle for an extended period. 0 No idle line detected. 1 Idle line was detected.
3 OR	Receiver Overrun Flag OR is set when a new serial character is ready to be transferred to the receive data buffer, but the previously received character has not been read from UART_D yet. In this case, the new character, and all associated error information, is lost because there is no room to move it into UART_D. To clear OR, write a logic 1 to the OR flag. 0 No overrun. 1 Receive overrun (new UART data lost).
2 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, the flag NF is set at the same time as RDRF is set for the character. To clear NF, write logic one to the NF. 0 No noise detected. 1 Noise detected in the received character in UART_D.
1 FE	Framing Error Flag FE is set at the same time as RDRF when the receiver detects a logic 0 where a stop bit was expected. This suggests the receiver was not properly aligned to a character frame. To clear FE, write a logic one to the FE flag. 0 No framing error detected. This does not guarantee the framing is correct. 1 Framing error.
0 PF	Parity Error Flag PF is set at the same time as RDRF 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.

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
	0 No parity error. 1 Parity error.

32.2.6 UART Status Register 2 (UARTx_S2)

This register contains one read-only status flag.

When using an internal oscillator in a LIN system, it is necessary to raise the break detection threshold one bit time. Under the worst case timing conditions allowed in LIN, it is possible that a 0x00 data character can appear to be 10.26 bit times long at a slave running 14% faster than the master. This would trigger normal break detection circuitry designed to detect a 10-bit break symbol. When the LBKDE bit is set, framing errors are inhibited and the break detection threshold increases, preventing false detection of a 0x00 data character as a LIN break symbol.

Address: Base address + 5h offset

Bit	7	6	5	4	3	2	1	0
Read	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_S2 field descriptions

Field	Description
7 LBKDIF	LIN Break Detect Interrupt Flag LBKDIF is set when the LIN break detect circuitry is enabled and a LIN break character is detected. LBKDIF is cleared by writing a 1 to it. 0 No LIN break character has been detected. 1 LIN break character has been detected.
6 RXEDGIF	UART_RX Pin Active Edge Interrupt Flag RXEDGIF is set when an active edge, falling if RXINV = 0, rising if RXINV=1, on the UART_RX pin occurs. RXEDGIF is cleared by writing a 1 to it. 0 No active edge on the receive pin has occurred. 1 An active edge on the receive pin has occurred.
5 MSBF	MSB First Setting this bit reverses the order of the bits that are transmitted and received on the wire. This bit does not affect the polarity of the bits, the location of the parity bit or the location of the start or stop bits. This bit should only be changed when the transmitter and receiver are both disabled.

Table continues on the next page...

UARTx_S2 field descriptions (continued)

Field	Description
	0 LSB (bit0) is the first bit that is transmitted following the start bit. Further, the first bit received after the start bit is identified as bit0. 1 MSB (bit9, bit8, bit7 or bit6) is the first bit that is transmitted following the start bit depending on the setting of C1[M], C1[PE] and C4[M10]. Further, the first bit received after the start bit is identified as bit9, bit8, bit7 or bit6 depending on the setting of C1[M] and C1[PE].
4 RXINV	Receive Data Inversion Setting this bit reverses the polarity of the received data input. **NOTE:** Setting RXINV inverts the UART_RXD input for all cases: data bits, start and stop bits, break, and idle. 0 Receive data not inverted. 1 Receive data inverted.
3 RWUID	Receive Wake Up Idle Detect RWUID controls whether the idle character that wakes up the receiver sets the IDLE bit. This bit should only be changed when the receiver is disabled. 0 During receive standby state (RWU = 1), the IDLE bit does not get set upon detection of an idle character. 1 During receive standby state (RWU = 1), the IDLE bit gets set upon detection of an idle character.
2 BRK13	Break Character Generation Length BRK13 selects a longer transmitted break character length. Detection of a framing error is not affected by the state of this bit. This bit should only be changed when the transmitter is disabled. 0 Break character is transmitted with length of 10 bit times (if M = 0, SBNS = 0) or 11 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 12 (if M = 1, SBNS = 1 or M10 = 1, SNBS = 0) or 13 (if M10 = 1, SNBS = 1). 1 Break character is transmitted with length of 13 bit times (if M = 0, SBNS = 0) or 14 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 15 (if M = 1, SBNS = 1 or M10 = 1, SNBS = 0) or 16 (if M10 = 1, SNBS = 1).
1 LBKDE	LIN Break Detection Enable LBKDE selects a longer break character detection length. While LBKDE is set, framing error (FE) and receive data register full (RDRF) flags are prevented from setting. 0 Break character is detected at length 10 bit times (if M = 0, SBNS = 0) or 11 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 12 (if M = 1, SBNS = 1 or M10 = 1, SNBS = 0) or 13 (if M10 = 1, SNBS = 1). 1 Break character is detected at length of 11 bit times (if M = 0, SBNS = 0) or 12 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 14 (if M = 1, SBNS = 1 or M10 = 1, SNBS = 0) or 15 (if M10 = 1, SNBS = 1).
0 RAF	Receiver Active Flag RAF is set when the UART receiver detects the beginning of a valid start bit, and RAF is cleared automatically when the receiver detects an idle line. 0 UART receiver idle waiting for a start bit. 1 UART receiver active (UART_RXD input not idle).

32.2.7 UART Control Register 3 (UARTx_C3)

Address: Base address + 6h offset

Bit	7	6	5	4	3	2	1	0
Read Write	R8T9	R9T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
Reset	0	0	0	0	0	0	0	0

UARTx_C3 field descriptions

Field	Description
7 R8T9	Receive Bit 8 / Transmit Bit 9 When the UART is configured for 9-bit data (M = 1), R8 can be thought of as a ninth receive data bit to the left of the msb of the buffered data in the UART_D register. When reading 9-bit data, read R8 before reading UART_D because reading UART_D completes automatic flag clearing sequences that could allow R8 and UART_D to be overwritten with new data. When the UART is configured for 10-bit data (M10 = 1), T9 may be thought of as a tenth transmit data bit. When writing 10-bit data, the entire 10-bit value is transferred to the UART transmit buffer when UART_D is written so T9 and T8 should be written, if it needs to change from its previous value, before UART_D is written. If T9 and T8 do not need to change in the new value, such as when it is used to generate mark or space parity, they need not be written each time UART_D is written.
6 R9T8	Receive Bit 9 / Transmit Bit 8 When the UART is configured for 9-bit data (M = 1), T8 may be thought of as a ninth transmit data bit to the left of the msb of the data in the UART_D register. When writing 9-bit data, the entire 9-bit value is transferred to the UART transmit buffer after UART_D is written so T8 should be written, if it needs to change from its previous value, before UART_D is written. If T8 does not need to change in the new value, such as when it is used to generate mark or space parity, it need not be written each time UART_D is written. When the UART is configured for 10-bit data (M10 = 1), R9 can be thought of as a tenth receive data bit. When reading 10-bit data, read R9 and R8 before reading UART_D because reading UART_D completes automatic flag clearing sequences that could allow R8, R9 and UART_D to be overwritten with new data.
5 TXDIR	UART_TX Pin Direction in Single-Wire Mode When the UART is configured for single-wire half-duplex operation (LOOPS = RSRC = 1), this bit determines the direction of data at the UART_TXD pin. When clearing TXDIR, the transmitter will finish receiving the current character (if any) before the receiver starts receiving data from the UART_TXD pin. 0 UART_TXD pin is an input in single-wire mode. 1 UART_TXD pin is an output in single-wire mode.
4 TXINV	Transmit Data Inversion Setting this bit reverses the polarity of the transmitted data output. **NOTE:** Setting TXINV inverts the UART_TXD output for all cases: data bits, start and stop bits, break, and idle. 0 Transmit data not inverted. 1 Transmit data inverted.
3 ORIE	Overrun Interrupt Enable This bit enables the overrun flag (OR) to generate hardware interrupt requests.

Table continues on the next page...

UARTx_C3 field descriptions (continued)

Field	Description
	0 OR interrupts disabled; use polling. 1 Hardware interrupt requested when OR is set.
2 NEIE	Noise Error Interrupt Enable This bit enables the noise flag (NF) to generate hardware interrupt requests. 0 NF interrupts disabled; use polling. 1 Hardware interrupt requested when NF is set.
1 FEIE	Framing Error Interrupt Enable This bit enables the framing error flag (FE) to generate hardware interrupt requests. 0 FE interrupts disabled; use polling. 1 Hardware interrupt requested when FE is set.
0 PEIE	Parity Error Interrupt Enable This bit enables the parity error flag (PF) to generate hardware interrupt requests. 0 PF interrupts disabled; use polling). 1 Hardware interrupt requested when PF is set.

32.2.8 UART Data Register (UARTx_D)

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 UART status flags.

Address: Base address + 7h offset

Bit	7	6	5	4	3	2	1	0
Read Write	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0
Reset	0	0	0	0	0	0	0	0

UARTx_D field descriptions

Field	Description
7 R7T7	Read receive data buffer 7 or write transmit data buffer 7.
6 R6T6	Read receive data buffer 6 or write transmit data buffer 6.
5 R5T5	Read receive data buffer 5 or write transmit data buffer 5.

Table continues on the next page...

UARTx_D field descriptions (continued)

Field	Description
4 R4T4	Read receive data buffer 4 or write transmit data buffer 4.
3 R3T3	Read receive data buffer 3 or write transmit data buffer 3.
2 R2T2	Read receive data buffer 2 or write transmit data buffer 2.
1 R1T1	Read receive data buffer 1 or write transmit data buffer 1.
0 R0T0	Read receive data buffer 0 or write transmit data buffer 0.

32.2.9 UART Match Address Registers 1 (UARTx_MA1)

The MA1 and MA2 registers are compared to input data addresses when the most significant bit is set and the associated C4[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 C4[MAEN] bit is clear.

Address: Base address + 8h offset

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

UARTx_MA1 field descriptions

Field	Description
7–0 MA	Match Address

32.2.10 UART Match Address Registers 2 (UARTx_MA2)

The MA1 and MA2 registers are compared to input data addresses when the most significant bit is set and the associated C4[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 C4[MAEN] bit is clear.

Address: Base address + 9h offset

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

UARTx_MA2 field descriptions

Field	Description
7–0 MA	Match Address

32.2.11 UART Control Register 4 (UARTx_C4)

Address: Base address + Ah offset

Bit	7	6	5	4	3	2	1	0
Read Write	MAEN1	MAEN2	M10	OSR				
Reset	0	0	0	0	1	1	1	1

UARTx_C4 field descriptions

Field	Description
7 MAEN1	Match Address Mode Enable 1 Refer to Match address operation for more information. 0 All data received is transferred to the data buffer if MAEN2 is cleared. 1 All data received with the most significant bit cleared, is discarded. All data received with the most significant bit set, is compared with contents of MA1 register. If no match occurs, the data is discarded. If match occurs, data is transferred to the data buffer.
6 MAEN2	Match Address Mode Enable 2 Refer to Match address operation for more information. 0 All data received is transferred to the data buffer if MAEN1 is cleared. 1 All data received with the most significant bit cleared, is discarded. All data received with the most significant bit set, is compared with contents of MA2 register. If no match occurs, the data is discarded. If match occurs, data is transferred to the data buffer.

Table continues on the next page...

UARTx_C4 field descriptions (continued)

Field	Description
5 M10	10-bit Mode select The M10 bit causes a tenth bit to be part of the serial transmission. This bit should only be changed when the transmitter and receiver are both disabled. 0 Receiver and transmitter use 8-bit or 9-bit data characters. 1 Receiver and transmitter use 10-bit data characters.
4–0 OSR	Over Sampling Ratio This field configures the oversampling ratio for the receiver between 4x (00011) and 32x (11111). Writing an invalid oversampling ratio will default to an oversampling ratio of 16 (01111). This field should only be changed when the transmitter and receiver are both disabled.

32.2.12 UART Control Register 5 (UARTx_C5)

Address: Base address + Bh offset

Bit	7	6	5	4	3	2	1	0
Read	TDMAE	0	RDMAE	0			BOTHEDGE	RESYNCDIS
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_C5 field descriptions

Field	Description
7 TDMAE	Transmitter DMA Enable TDMAE configures the transmit data register empty flag, S1[TDRE], to generate a DMA request. 0 DMA request disabled. 1 DMA request enabled.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 RDMAE	Receiver Full DMA Enable RDMAE configures the receiver data register full flag, S1[RDRF], to generate a DMA request. 0 DMA request disabled. 1 DMA request enabled.
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 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.

Table continues on the next page...

UARTx_C5 field descriptions (continued)

Field	Description
	0 Receiver samples input data using the rising edge of the baud rate clock. 1 Receiver samples input data using the rising and falling edge of the baud rate clock.
0 RESYNCDIS	Resynchronization Disable When set, disables the resynchronization of the received data word when a data one followed by data zero transition is detected. This bit should only be changed when the receiver is disabled. 0 Resynchronization during received data word is supported 1 Resynchronization during received data word is disabled

32.3 Functional description

The UART 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 UART.

32.3.1 Baud rate generation

A 13-bit modulus counter in the baud rate generator derive the baud rate for both the receiver and the transmitter.

The value from 1 to 8191 written to SBR[12:0] determines the baud clock divisor for the asynchronous UART baud clock. The SBR bits are in the UART baud rate registers, BDH and BDL. The baud rate clock drives the receiver, while the transmitter is driven by the baud rate clock divided by the over sampling ratio. Depending on the over sampling ratio, the receiver has an acquisition rate of 4 to 32 samples per bit time.

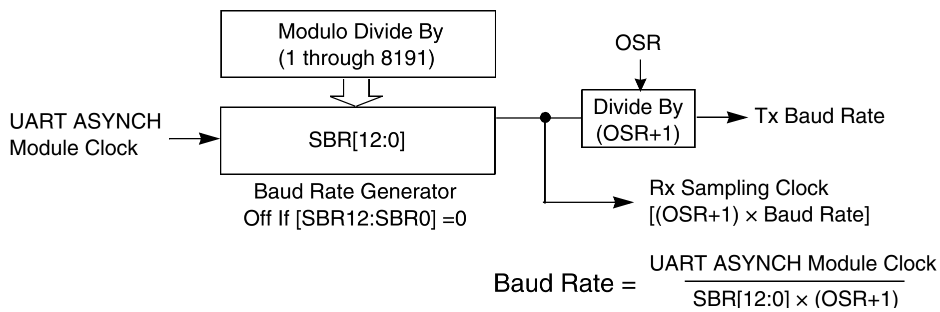


Figure 32-27. UART baud rate generation

Baud rate generation is subject to two sources of error:

- Integer division of the module clock may not give the exact target frequency.
- Synchronization with the asynchronous UART baud clock can cause phase shift.

32.3.2 Transmitter functional description

This section describes the overall block diagram for the UART transmitter, as well as specialized functions for sending break and idle characters.

The transmitter output (UART_TX) idle state defaults to logic high, C3[TXINV] is cleared following reset. The transmitter output is inverted by setting C3[TXINV]. The transmitter is enabled by setting the C2[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 UART data register.

The central element of the UART transmitter is the transmit shift register that is 10-bit to 13 bits long depending on the setting in the C1[M], C2[M10] and BDH[SBNS] control bits. For the remainder of this section, assume C1[M], C2[M10] and BDH[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 UART 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 (S1[TDRE]) status flag is set to indicate another character may be written to the transmit data buffer at UART_D.

If no new character is waiting in the transmit data buffer after a stop bit is shifted out the UART_TX pin, the transmitter sets the transmit complete flag and enters an idle mode, with UART_TX high, waiting for more characters to transmit.

Writing 0 to C2[TE] does not immediately disable the transmitter. The current transmit activity in progress must first be completed. This includes data characters in progress, queued idle characters, and queued break characters.

32.3.2.1 Send break and queued idle

The UART_C2[SBK] bit sends break characters originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0, 10-bit to 12-bit times including the start and stop bits. A longer break of 13-bit times can be enabled by setting UART_S2[BRK13]. Normally, a program would wait for UART_S1[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, write 1, and then write 0 to the UART_C2[SBK] bit. This action queues a break character to be sent as soon as the shifter is available. If UART_C2[SBK] remains 1 when the queued break moves into the shifter, synchronized to the baud rate clock, an additional break character is queued. If the receiving device is another Freescale Semiconductor UART, the break characters are received as 0s in all data bits and a framing error (UART_S1[FE] = 1) occurs.

When idle-line wakeup is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for UART_S1[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the UART_C2[TE] bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish whileUART_C2[TE] is cleared, the UART transmitter never actually releases control of the UART_TX pin.

The length of the break character is affected by the UART_S2[BRK13], UART_C1[M] and UART_C4[M10] bits as shown below.

Table 32-27. Break character length

BRK13	M	M10	SBNS	Break character length
0	0	0	0	10 bit times
0	0	0	1	11 bit times
0	1	0	0	11 bit times
0	1	0	1	12 bit times
0	X	1	0	12 bit times
0	X	1	1	13 bit times
1	0	0	0	13 bit times
1	0	0	1	14 bit times
1	1	0	0	14 bit times
1	1	0	1	15 bit times
1	X	1	0	15 bit times
1	X	1	1	16 bit times

32.3.3 Receiver functional description

In this section, the receiver block diagram is a guide for the overall receiver functional description.

Next, the data sampling technique used to reconstruct receiver data is described in more detail. Finally, two variations of the receiver wakeup function are explained.

The receiver input is inverted by setting UART_S2[RXINV]. The receiver is enabled by setting the UART_C2[RE] bit. Character frames consist of a start bit of logic 0, eight to ten data bits (msb or lsb first), and one or two stop bits of logic 1. For information about 9-bit or 10-bit data mode, refer to 8-bit, 9-bit and 10-bit data modes. For the remainder of this discussion, assume the UART 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 (UART_S1[RDRF]) status flag is set. If UART_S1[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 UART receiver is double-buffered, the program has one full character time after UART_S1[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 (UART_S1[RDRF] = 1), it gets the data from the receive data register by reading UART_D. Refer to Interrupts and status flags for details about flag clearing.

32.3.3.1 Data sampling technique

The UART receiver supports an 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 UART_RX serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The oversampling baud rate clock divides the bit time into 4 to 32 segments from 1 to OSR (where OSR is the configured oversampling ratio). When a falling edge is located, three more samples are taken at (OSR/2), (OSR/2)+1, and (OSR/2)+2 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character. If another falling edge is detected before the receiver is considered synchronized, the receiver restarts the sampling from the first segment.

The receiver then samples each bit time, including the start and stop bits, at (OSR/2), (OSR/2)+1, and (OSR/2)+2 to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. If any sample in any bit time, including the start and stop bits, in a character frame fails to agree with the logic level for that bit, the noise flag (UART_S1[NF]) is set when the received character is transferred to the receive data buffer.

When the UART 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.

32.3.3.2 Receiver wakeup operation

Receiver wakeup is a hardware mechanism that allows an UART receiver to ignore the characters in a message intended for a different UART receiver. In such a system, 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(UART_C2[RWU]). When RWU bit is set, the status flags associated with the receiver, with the exception of the idle bit, IDLE, when UART_S2[RWUID] bit is set, 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 UART_C2[RWU] to 0 so all receivers wake up in time to look at the first character(s) of the next message.

32.3.3.2.1 Idle-line wakeup

When wake is cleared, the receiver is configured for idle-line wakeup. In this mode, UART_C2[RWU] is cleared automatically when the receiver detects a full character time of the idle-line level. The UART_C1[M] and UART_C4[M10] control bit selects 8-bit to 10-bit data mode and the UART_BDH[SBNS] bit selects 1-bit or 2-bit stop bit number that determines how many bit times of idle are needed to constitute a full character time, 10 to 13 bit times because of the start and stop bits.

When UART_C2[RWU] is one and UART_S2[RWUID] is zero, the idle condition that wakes up the receiver does not set the UART_S1[IDLE] flag. The receiver wakes up and waits for the first data character of the next message that sets the UART_S1[RDRF] flag and generates an interrupt if enabled. When UART_S2[RWUID] is one, any idle condition sets the UART_S1[IDLE] flag and generates an interrupt if enabled, regardless of whether UART_C2[RWU] is zero or one.

The idle-line type (UART_C1[ILT]) control bit selects one of two ways to detect an idle line. When UART_C1[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 UART_C1[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.

32.3.3.2.2 Address-mark wakeup

When wake is set, the receiver is configured for address-mark wakeup. In this mode, UART_C2[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 UART_C2[RWU] bit before the stop bits are received and sets the UART_S1[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.

32.3.3.2.3 Match address operation

Match address operation is enabled when the UART_C4[MAEN1] or UART_C4[MAEN2] bit is set. In this function, a frame received by the UART_RX pin with a logic 1 in the bit position immediately preceding the stop bit is considered an address and is compared with the associated MA1 or MA2 register. The frame is only transferred to the receive buffer, and UART_S1[RDRF] is set, if the comparison matches. All subsequent frames 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 frames with logic zero in the bit position immediately preceding the stop bit are also discarded. If both the UART_C4[MAEN1] and UART_C4[MAEN2] bits are negated, the receiver operates normally and all data received is transferred to the receive data buffer.

Match Address operation functions in the same way for both MA1 and MA2 registers.

- If only one of UART_C4[MAEN1] and UART_C4[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 UART_C4[MAEN1] and UART_C4[MAEN2] are asserted, a marked address is compared with both match registers and data is transferred only on a match with either register.

32.3.4 Additional UART functions

The following sections describe additional UART functions.

32.3.4.1 8-bit, 9-bit and 10-bit data modes

The UART system, transmitter and receiver, can be configured to operate in 9-bit data mode by setting the UART_C1[M] or 10-bit data mode by setting UART_C4[M10]. In 9-bit mode, there is a ninth data bit to the left of the msb of the UART data register, in 10-bit mode there is a tenth data bit. For the transmit data buffer, these bits are stored in T8 and T9 in UART_C3. For the receiver, these bits are held in UART_C3[R8] and UART_C3[R9].

For coherent writes to the transmit data buffer, write to UART_C3[T8] and UART_C3[T9] before writing to UART_D.

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 T8 and T9 again. When data is transferred from the transmit data buffer to the transmit shifter, the value in T8 and T9 is copied at the same time data is transferred from UART_D 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.

32.3.4.2 Loop mode

When UART_C1[LOOPS] is set, the UART_C1[RSRC] bit in the same register chooses between loop mode (UART_C1[RSRC] = 0) or single-wire mode (UART_C1[RSRC] = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and the UART_RX pin is not used by the UART.

32.3.4.3 Single-wire operation

When UART_C1[LOOPS] is set, the RSRC bit in the same register chooses between loop mode (UART_C1[RSRC] = 0) or single-wire mode (UART_C1[RSRC] = 1). Single-wire mode implements a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the UART_TX pin (the UART_RX pin is not used).

In single-wire mode, the UART_C3[TXDIR] bit controls the direction of serial data on the UART_TX pin. When UART_C3[TXDIR] is cleared, the UART_TX pin is an input to the UART receiver and the transmitter is temporarily disconnected from the UART_TX pin so an external device can send serial data to the receiver. When UART_C3[TXDIR] is set, the UART_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.

32.3.5 Interrupts and status flags

The UART system generates three separate interrupts to reduce the amount of software needed to isolate the cause of the interrupt.

One interrupt is associated with the transmitter for TDRE and TC events. Another interrupt is associated with the receiver for RDRF, IDLE, RXEDGIF, and LBKDIF events. A third interrupt is used for OR, NF, FE, and PF error conditions. Each of these ten interrupt sources can be separately masked by local interrupt enable masks. The flags can be polled by software when the local masks are cleared to disable generation of hardware interrupt requests.

The UART transmitter has two status flags that can optionally generate hardware interrupt requests. Transmit data register empty (UART_S1[TDRE]) indicates when there is room in the transmit data buffer to write another transmit character to UART_D. If the transmit interrupt enable (UART_C2[TIE]) bit is set, a hardware interrupt is requested when UART_S1[TDRE] is set. Transmit complete (UART_S1[TC]) indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with UART_TX at the inactive level. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (UART_C2[TCIE]) bit is set, a hardware interrupt is requested when UART_S1[TC] is set. Instead of hardware interrupts, software polling may be used to monitor the UART_S1[TDRE] and UART_S1[TC] status flags if the corresponding UART_C2[TIE] or UART_C2[TCIE] local interrupt masks are cleared.

When a program detects that the receive data register is full (UART_S1[RDRF] = 1), it gets the data from the receive data register by reading UART_D. The UART_S1[RDRF] flag is cleared by reading UART_D.

The IDLE status flag includes logic that prevents it from getting set repeatedly when the UART_RX line remains idle for an extended period of time. IDLE is cleared by writing 1 to the UART_S1[IDLE] flag. After UART_S1[IDLE] has been cleared, it cannot become set again until the receiver has received at least one new character and has set UART_S1[RDRF].

If the associated error was detected in the received character that caused UART_S1[RDRF] to be set, the error flags - noise flag (UART_S1[NF]), framing error (UART_S1[FE]), and parity error flag (UART_S1[PF]) - are set at the same time as UART_S1[RDRF]. These flags are not set in overrun cases.

If UART_S1[RDRF] was already set when a new character is ready to be transferred from the receive shifter to the receive data buffer, the overrun (UART_S1[OR]) flag is set instead of the data along with any associated NF, FE, or PF condition is lost.

At any time, an active edge on the UART_RX serial data input pin causes the UART_S2[RXEDGIF] flag to set. The UART_S2[RXEDGIF] flag is cleared by writing a 1 to it. This function depends on the receiver being enabled (UART_C2[RE] = 1).

Chapter 33
Universal Asynchronous Receiver/Transmitter (UART)

33.1 Introduction

33.1.1 Features

Features of UART module include:

- Full-duplex, standard non-return-to-zero (NRZ) format
- Double-buffered transmitter and receiver with separate enables
- Programmable baud rates (13-bit modulo divider)
- Interrupt-driven 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
- Hardware parity generation and checking
- Programmable 8-bit or 9-bit character length
- Programmable 1-bit or 2-bit stop bits
- Receiver wakeup by idle-line or address-mark
- Optional 13-bit break character generation / 11-bit break character detection
- Selectable transmitter output polarity
- 5-channel DMA interface

33.1.2 Modes of operation

See Section Functional description for details concerning UART operation in these modes:

- 8- and 9-bit data modes
- Stop mode operation
- Loop mode
- Single-wire mode

33.1.3 Block diagram

The following figure shows the transmitter portion of the UART.

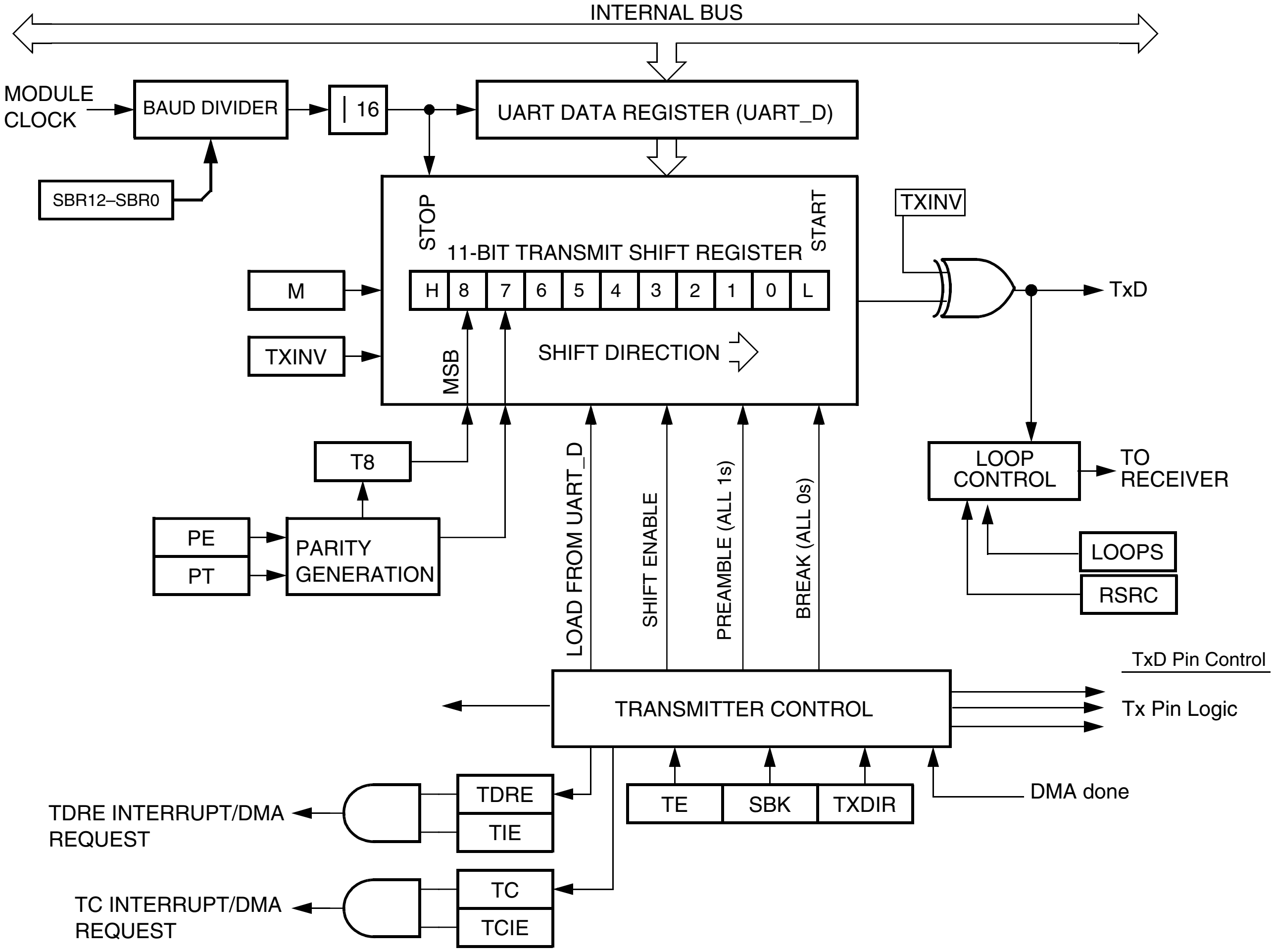


Figure 33-1. UART transmitter block diagram

The following figure shows the receiver portion of the UART.

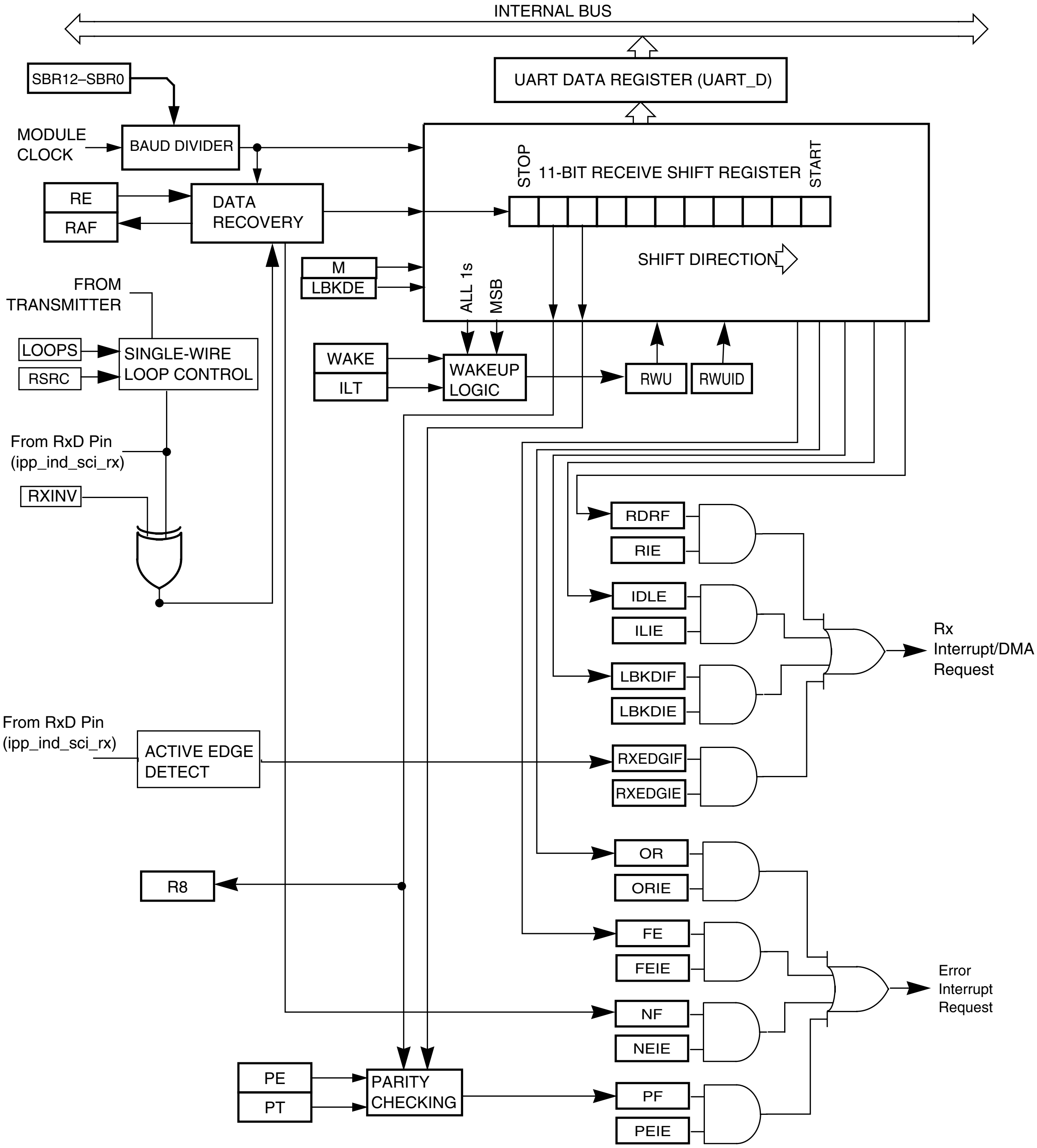


Figure 33-2. UART receiver block diagram

33.2 UART signal descriptions

The UART signals are shown in the table found here.

Table 33-1. UART signal descriptions

Signal	Description	I/O
RxD	Receive data	I
TxD	Transmit data	I/O

33.2.1 Detailed signal descriptions

The detailed signal descriptions of the UART are shown in the following table.

Table 33-2. UART—Detailed signal descriptions

Signal	I/O	Description	
RxD	I	Receive data. Serial data input to receiver.	
		State meaning	Whether RxD is interpreted as a 1 or 0 depends on the bit encoding method along with other configuration settings.
		Timing	Sampled at a frequency determined by the module clock divided by the baud rate.
TxD	I/O	Transmit data. Serial data output from transmitter.	
		State meaning	Whether TxD is interpreted as a 1 or 0 depends on the bit encoding method along with other configuration settings.
		Timing	Driven at the beginning or within a bit time according to the bit encoding method along with other configuration settings. Otherwise, transmissions are independent of reception timing.

33.3 Register definition

The UART has 8-bit registers to control baud rate, select UART options, report UART status, select DMA options, and for transmit/receive data.

Refer to the direct-page register summary in the memory chapter of this document or the absolute address assignments for all UART registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_B000	UART Baud Rate Register: High (UART1_BDH)	8	R/W	00h	33.3.1/639
4006_B001	UART Baud Rate Register: Low (UART1_BDL)	8	R/W	04h	33.3.2/640
4006_B002	UART Control Register 1 (UART1_C1)	8	R/W	00h	33.3.3/641
4006_B003	UART Control Register 2 (UART1_C2)	8	R/W	00h	33.3.4/642
4006_B004	UART Status Register 1 (UART1_S1)	8	R	C0h	33.3.5/644
4006_B005	UART Status Register 2 (UART1_S2)	8	R/W	00h	33.3.6/645
4006_B006	UART Control Register 3 (UART1_C3)	8	R/W	00h	33.3.7/647
4006_B007	UART Data Register (UART1_D)	8	R/W	00h	33.3.8/648
4006_B008	UART Control Register 4 (UART1_C4)	8	R/W	00h	33.3.9/649
4006_C000	UART Baud Rate Register: High (UART2_BDH)	8	R/W	00h	33.3.1/639
4006_C001	UART Baud Rate Register: Low (UART2_BDL)	8	R/W	04h	33.3.2/640
4006_C002	UART Control Register 1 (UART2_C1)	8	R/W	00h	33.3.3/641
4006_C003	UART Control Register 2 (UART2_C2)	8	R/W	00h	33.3.4/642
4006_C004	UART Status Register 1 (UART2_S1)	8	R	C0h	33.3.5/644
4006_C005	UART Status Register 2 (UART2_S2)	8	R/W	00h	33.3.6/645
4006_C006	UART Control Register 3 (UART2_C3)	8	R/W	00h	33.3.7/647
4006_C007	UART Data Register (UART2_D)	8	R/W	00h	33.3.8/648
4006_C008	UART Control Register 4 (UART2_C4)	8	R/W	00h	33.3.9/649

33.3.1 UART Baud Rate Register: High (UARTx_BDH)

This register, along with UART_BDL, controls the prescale divisor for UART baud rate generation. To update the 13-bit baud rate setting SBR[12:0], first write to UART_BDH to buffer the high half of the new value and then write to UART_BDL. The working value in UART_BDH does not change until UART_BDL is written.

Address: Base address + h offset

Bit	7	6	5	4	3	2	1	0
Read/Write	LBKDIE	RXEDGIE	SBNS	SBR				
Reset	0	0	0	0	0	0	0	0

UARTx_BDH field descriptions

Field	Description
7 LBKDIE	LIN Break Detect Interrupt Enable (for LBKDIF)

Table continues on the next page...

UARTx_BDH field descriptions (continued)

Field	Description
	0 Hardware interrupts from UART_S2[LBKDIF] disabled (use polling). 1 Hardware interrupt requested when UART_S2[LBKDIF] flag is 1.
6 RXEDGIE	RxD Input Active Edge Interrupt Enable (for RXEDGIF) 0 Hardware interrupts from UART_S2[RXEDGIF] disabled (use polling). 1 Hardware interrupt requested when UART_S2[RXEDGIF] flag is 1.
5 SBNS	Stop Bit Number Select SBNS determines whether data characters are one or two stop bits. 0 One stop bit. 1 Two stop bit.
4–0 SBR	Baud Rate Modulo Divisor. The 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the UART baud rate generator. When BR is cleared, the UART baud rate generator is disabled to reduce supply current. When BR is 1 - 8191, the UART baud rate equals BUSCLK/(16×BR).

33.3.2 UART Baud Rate Register: Low (UARTx_BDL)

This register, along with UART_BDH, control the prescale divisor for UART baud rate generation. To update the 13-bit baud rate setting [SBR12:SBR0], first write to UART_BDH to buffer the high half of the new value and then write to UART_BDL. The working value in UART_BDH does not change until UART_BDL is written.

UART_BDL is reset to a non-zero value, so after reset, the baud rate generator remains disabled until the first time the receiver or transmitter is enabled; that is, 1 is written to UART_C2[RE] or UART_C2[TE].

Address: Base address + h offset

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

UARTx_BDL field descriptions

Field	Description
7–0 SBR	Baud Rate Modulo Divisor These 13 bits in SBR[12:0] are referred to collectively as BR, which set the modulo divide rate for the UART baud rate generator. When BR is cleared, the UART baud rate generator is disabled to reduce supply current. When BR is 1 - 8191, the UART baud rate equals BUSCLK/(16×BR).

33.3.3 UART Control Register 1 (UARTx_C1)

This read/write register controls various optional features of the UART system.

Address: Base address + h offset

Bit	7	6	5	4	3	2	1	0
Read Write	LOOPS	UARTSWAI	RSRC	M	WAKE	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0

UARTx_C1 field descriptions

Field	Description
7 LOOPS	Loop Mode Select Selects between loop mode and normal 2-pin full-duplex modes. When LOOPS is set, the transmitter output is internally connected to the receiver input. 0 Normal operation - RxD and TxD use separate pins. 1 Loop mode or single-wire mode where transmitter outputs are internally connected to receiver input. (See RSRC bit.) RxD pin is not used by UART.
6 UARTSWAI	UART Stops in Wait Mode 0 UART clocks continue to run in Wait mode so the UART can be the source of an interrupt that wakes up the CPU. 1 UART clocks freeze while CPU is in Wait mode.
5 RSRC	Receiver Source Select This field has no meaning or effect unless LOOPS is set to 1. When LOOPS is set, the receiver input is internally connected to the TxD pin and RSRC determines whether this connection is also connected to the transmitter output. 0 Provided LOOPS is set, RSRC is cleared, selects internal loop back mode and the UART does not use the RxD pins. 1 Single-wire UART mode where the TxD pin is connected to the transmitter output and receiver input.
4 M	9-Bit or 8-Bit Mode Select This field configures the UART to be operated in 9-bit or 8-bit data mode. 0 Normal - start + 8 data bits (lsb first) + stop. 1 Receiver and transmitter use 9-bit data characters start + 8 data bits (lsb first) + 9th data bit + stop.
3 WAKE	Receiver Wakeup Method Select This field selects the receiver wakeup method. 0 Idle-line wake-up. 1 Address-mark wake-up.
2 ILT	Idle Line Type Select

Table continues on the next page...

UARTx_C1 field descriptions (continued)

Field	Description
	Setting this field to 1 ensures that the stop bits and logic 1 bits at the end of a character do not count toward the 10 or 11 bit times of logic high level needed by the idle line detection logic. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.
1 PE	Parity Enable Enables hardware parity generation and checking. When parity is enabled, the most significant bit (msb) of the data character, eighth or ninth data bit, is treated as the parity bit. 0 No hardware parity generation or checking. 1 Parity enabled.
0 PT	Parity Type Provided parity is enabled (PE = 1), this field selects even or odd parity. Odd parity means the total number of 1s in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even. 0 Even parity. 1 Odd parity.

33.3.4 UART Control Register 2 (UARTx_C2)

This register can be read or written at any time.

Address: Base address + h offset

Bit	7	6	5	4	3	2	1	0
Read Write	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0

UARTx_C2 field descriptions

Field	Description
7 TIE	Transmit Interrupt Enable for TDRE 0 Hardware interrupts from TDRE disabled; use polling. 1 Hardware interrupt requested when TDRE flag is 1.
6 TCIE	Transmission Complete Interrupt Enable for TC 0 Hardware interrupts from TC disabled; use polling. 1 Hardware interrupt requested when TC flag is 1.
5 RIE	Receiver Interrupt Enable for RDRF 0 Hardware interrupts from S1[RDRF] disabled; use polling. 1 Hardware interrupt requested when S1[RDRF] flag is 1.
4 ILIE	Idle Line Interrupt Enable for IDLE

Table continues on the next page...

UARTx_C2 field descriptions (continued)

Field	Description
	0 Hardware interrupts from S1[IDLE] disabled; use polling. 1 Hardware interrupt requested when S1[IDLE] flag is 1.
3 TE	Transmitter Enable TE must be 1 to use the UART transmitter. When TE is set, the UART forces the TxD pin to act as an output for the UART system. When the UART is configured for single-wire operation (LOOPS = RSRC = 1), TXDIR controls the direction of traffic on the single UART communication line (TxD pin). TE can also queue an idle character by clearing TE and then setting TE while a transmission is in progress. When 0 is written to TE, the transmitter keeps control of the port TxD pin until any data, queued idle, or queued break character finishes transmitting before allowing the pin to revert to a general-purpose I/O pin. 0 Transmitter off. 1 Transmitter on.
2 RE	Receiver Enable When the UART receiver is off, the RxD pin reverts to being a general-purpose port I/O pin. If C1[LOOPS] is set, the RxD pin reverts to being a general-purpose I/O pin even if RE is set. 0 Receiver off. 1 Receiver on.
1 RWU	Receiver Wakeup Control A 1 can be written to this field to place the UART receiver in a standby state where it waits for automatic hardware detection of a selected wake-up condition. The wake-up condition is an idle line between messages, WAKE = 0, idle-line wake-up, or a logic 1 in the most significant data bit in a character, WAKE = 1, address-mark wake-up. Application software sets RWU and, normally, a selected hardware condition automatically clears RWU. 0 Normal UART receiver operation. 1 UART receiver in standby waiting for wake-up condition.
0 SBK	Send Break Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 or 11 or 12, 13 or 14 or 15 if BRK13 = 1, bit times of logic 0 are queued as long as SBK is set. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. 0 Normal transmitter operation. 1 Queue break character(s) to be sent.

33.3.5 UART Status Register 1 (UARTx_S1)

This register has eight read-only status flags. Writes have no effect. Special software sequences, which do not involve writing to this register, clear these status flags.

Address: Base address + h offset

Bit	7	6	5	4	3	2	1	0
Read	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
Write								
Reset	1	1	0	0	0	0	0	0

UARTx_S1 field descriptions

Field	Description
7 TDRE	Transmit Data Register Empty Flag TDRE is set out of reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read UART_S1 with TDRE set and then write to the UART data register (UART_D). 0 Transmit data register (buffer) full. 1 Transmit data register (buffer) empty.
6 TC	Transmission Complete Flag TC is set out of reset and when TDRE is set and no data, preamble, or break character is being transmitted. TC is cleared automatically by reading UART_S1 with TC set and then executing one of the following operations: • Write to the UART data register (UART_D) to transmit new data • Queue a preamble by changing TE from 0 to 1 • Queue a break character by writing 1 to UART_C2[SBK] 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete).
5 RDRF	Receive Data Register Full Flag RDRF becomes set when a character transfers from the receive shifter into the receive data register (UART_D). To clear RDRF, read UART_S1 with RDRF set and then read the UART data register (UART_D). 0 Receive data register empty. 1 Receive data register full.
4 IDLE	Idle Line Flag IDLE is set when the UART receive line becomes idle for a full character time after a period of activity. When C1[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 or 11 bit times depending on the M control bit, needed for the receiver to detect an idle line. When ILT is set, the receiver doesn't start counting idle bit times until 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.

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
	To clear IDLE, read UART_S1 with IDLE set and then read the UART data register (UART_D). After IDLE has been cleared, it cannot become set again until after a new character has been received and RDRF has been set. IDLE is set only once even if the receive line remains idle for an extended period. 0 No idle line detected. 1 Idle line was detected.
3 OR	Receiver Overrun Flag OR is set when a new serial character is ready to be transferred to the receive data register (buffer), but the previously received character has not been read from UART_D yet. In this case, the new character, and all associated error information, is lost because there is no room to move it into UART_D. To clear OR, read UART_S1 with OR set and then read the UART data register (UART_D). 0 No overrun. 1 Receive overrun (new UART data lost).
2 NF	Noise Flag The advanced sampling technique used in the receiver takes seven samples during the start bit and three samples in each data bit and the stop bits. If any of these samples disagrees with the rest of the samples within any bit time in the frame, the flag NF is set at the same time as RDRF is set for the character. To clear NF, read UART_S1 and then read the UART data register (UART_D). 0 No noise detected. 1 Noise detected in the received character in UART_D.
1 FE	Framing Error Flag FE is set at the same time as RDRF when the receiver detects a logic 0 where the stop bits were expected. This suggests the receiver was not properly aligned to a character frame. To clear FE, read UART_S1 with FE set and then read the UART data register (UART_D). 0 No framing error detected. This does not guarantee the framing is correct. 1 Framing error.
0 PF	Parity Error Flag PF is set at the same time as RDRF 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, read UART_S1 and then read the UART data register (UART_D). 0 No parity error. 1 Parity error.

33.3.6 UART Status Register 2 (UARTx_S2)

This register contains one read-only status flag.

When using an internal oscillator in a LIN system, it is necessary to raise the break detection threshold one bit time. Under the worst case timing conditions allowed in LIN, it is possible that a 0x00 data character can appear to be 10.26 bit times long at a slave running 14% faster than the master. This would trigger normal break detection circuitry designed to detect a 10-bit break symbol. When the LBKDE bit is set, framing errors are inhibited and the break detection threshold changes from 10 bits to 11 bits, preventing false detection of a 0x00 data character as a LIN break symbol.

Address: Base address + h offset

Bit	7	6	5	4	3	2	1	0
Read	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_S2 field descriptions

Field	Description
7 LBKDIF	LIN Break Detect Interrupt Flag LBKDIF is set when the LIN break detect circuitry is enabled and a LIN break character is detected. LBKDIF is cleared by writing a 1 to it. 0 No LIN break character has been detected. 1 LIN break character has been detected.
6 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. 0 No active edge on the receive pin has occurred. 1 An active edge on the receive pin has occurred.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 RXINV	Receive Data Inversion Setting this field 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. 0 Receive data not inverted. 1 Receive data inverted.
3 RWUID	Receive Wake Up Idle Detect RWUID controls whether the idle character that wakes up the receiver sets S1[IDLE]. 0 During receive standby state (RWU = 1), S1[IDLE] does not get set upon detection of an idle character. 1 During receive standby state (RWU = 1), S1[IDLE] gets set upon detection of an idle character.
2 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 field. 0 Break character is transmitted with length of 10 bit times (if M = 0, SBNS = 0) or 11 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 12 (if M = 1, SBNS = 1). 1 Break character is transmitted with length of 13 bit times (if M = 0, SBNS = 0) or 14 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 15 (if M = 1, SBNS = 1).

Table continues on the next page...

UARTx_S2 field descriptions (continued)

Field	Description
1 LBKDE	LIN Break Detection Enable LBKDE enables the break detection. While LBKDE is set, S1[FE] and S1[RDRF] flags are prevented from setting. 0 Break detection is disabled. 1 Break detection is enabled (Break character is detected at length 11 bit times (if C1[M] = 0, BDH[SBNS] = 0) or 12 (if C1[M] = 1, BDH[SBNS] = 0 or C1[M] = 0, BDH[SBNS] = 1) or 13 (if C1[M] = 1, BDH[SBNS] = 1)).
0 RAF	Receiver Active Flag RAF is set when the UART receiver detects the beginning of a valid start bit, and RAF is cleared automatically when the receiver detects an idle line. This status flag can be used to check whether an UART character is being received before instructing the MCU to go to stop mode. 0 UART receiver idle waiting for a start bit. 1 UART receiver active (RxD input not idle).

33.3.7 UART Control Register 3 (UARTx_C3)

Address: Base address + h offset

Bit	7	6	5	4	3	2	1	0
Read	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_C3 field descriptions

Field	Description
7 R8	Ninth Data Bit for Receiver When the UART is configured for 9-bit data (C1[M] = 1), R8 can be thought of as a ninth receive data bit to the left of the msb of the buffered data in the UART_D register. When reading 9-bit data, read R8 before reading UART_D because reading UART_D completes automatic flag clearing sequences that could allow R8 and UART_D to be overwritten with new data.
6 T8	Ninth Data Bit for Transmitter When the UART is configured for 9-bit data (C1[M] = 1), T8 may be thought of as a ninth transmit data bit to the left of the msb of the data in the UART_D register. When writing 9-bit data, the entire 9-bit value is transferred to the UART shift register after UART_D is written so T8 should be written, if it needs to change from its previous value, before UART_D is written. If T8 does not need to change in the new value, such as when it is used to generate mark or space parity, it need not be written each time UART_D is written.
5 TXDIR	TxD Pin Direction in Single-Wire Mode When the UART is configured for single-wire half-duplex operation (LOOPS = RSRC = 1), this field determines the direction of data at the TxD pin.

Table continues on the next page...

UARTx_C3 field descriptions (continued)

Field	Description
	0 TxD pin is an input in single-wire mode. 1 TxD pin is an output in single-wire mode.
4 TXINV	Transmit Data Inversion Setting this field 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. 0 Transmit data not inverted. 1 Transmit data inverted.
3 ORIE	Overrun Interrupt Enable Enables the overrun flag (OR) to generate hardware interrupt requests. 0 OR interrupts disabled; use polling. 1 Hardware interrupt requested when OR is set.
2 NEIE	Noise Error Interrupt Enable Enables the noise flag (NF) to generate hardware interrupt requests. 0 NF interrupts disabled; use polling). 1 Hardware interrupt requested when NF is set.
1 FEIE	Framing Error Interrupt Enable Enables the framing error flag (FE) to generate hardware interrupt requests. 0 FE interrupts disabled; use polling). 1 Hardware interrupt requested when FE is set.
0 PEIE	Parity Error Interrupt Enable Enables the parity error flag (PF) to generate hardware interrupt requests. 0 PF interrupts disabled; use polling). 1 Hardware interrupt requested when PF is set.

33.3.8 UART Data Register (UARTx_D)

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 the UART status flags.

Address: Base address + h offset

Bit	7	6	5	4	3	2	1	0
Read Write	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0
Reset	0	0	0	0	0	0	0	0

UARTx_D field descriptions

Field	Description
7 R7T7	Read receive data buffer 7 or write transmit data buffer 7.
6 R6T6	Read receive data buffer 6 or write transmit data buffer 6.
5 R5T5	Read receive data buffer 5 or write transmit data buffer 5.
4 R4T4	Read receive data buffer 4 or write transmit data buffer 4.
3 R3T3	Read receive data buffer 3 or write transmit data buffer 3.
2 R2T2	Read receive data buffer 2 or write transmit data buffer 2.
1 R1T1	Read receive data buffer 1 or write transmit data buffer 1.
0 R0T0	Read receive data buffer 0 or write transmit data buffer 0.

33.3.9 UART Control Register 4 (UARTx_C4)

Address: Base address + h offset

Bit	7	6	5	4	3	2	1	0
Read	TDMAS	0	RDMAS	0	0		0	
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_C4 field descriptions

Field	Description
7 TDMAS	Transmitter DMA Select TDMAS configures the transmit data register empty flag, TDRE, to generate interrupt or DMA requests if TIE is set. **NOTE:** If UART_C2[TIE] is cleared, TDRE DMA and TDRE interrupt request signals are not asserted when the TDRE flag is set, regardless of the state of TDMAS. If UART_C2[TIE] and TDMAS are both set, then UART_C2[TCIE] must be cleared, and UART_D must not be written outside of servicing of a DMA request. 0 If TIE is set and the TDRE flag is set, the TDRE interrupt request signal is asserted to request interrupt service. 1 If TIE is set and the TDRE flag is set, the TDRE DMA request signal is asserted to request a DMA transfer.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 RDMAS	Receiver Full DMA Select

Table continues on the next page...

UARTx_C4 field descriptions (continued)

Field	Description
	RDMAS configures the receiver data register full flag, RDRF, to generate interrupt or DMA requests if RIE is set. **NOTE:** If RIE is cleared, the RDRF DMA and RDRF interrupt request signals are not asserted when the RDRF flag is set, regardless of the state of RDMAS. 0 If RIE is set and the RDRF flag is set, the RDRF interrupt request signal is asserted to request interrupt service. 1 If RIE is set and the RDRF flag is set, the RDRF DMA request signal is asserted to request a DMA transfer.
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.
2–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

33.4 Functional description

The UART allows full-duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other MCUs.

The UART comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. During normal operation, the MCU monitors the status of the UART, writes the data to be transmitted, and processes received data. The following describes each of the blocks of the UART.

33.4.1 Baud rate generation

As shown in the figure found here, the clock source for the UART baud rate generator is the bus-rate clock.

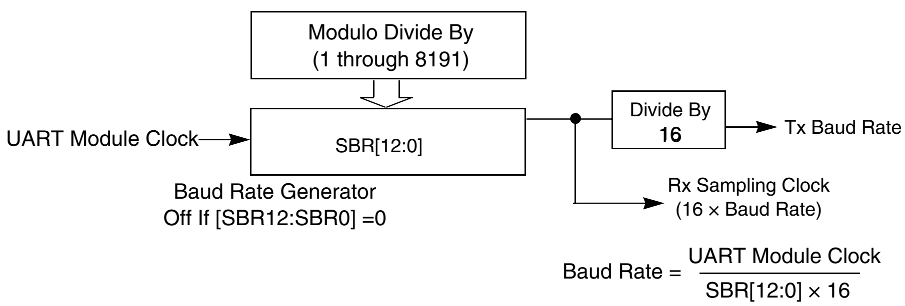


Figure 33-30. UART baud rate generation

UART communications require the transmitter and receiver, which typically derive baud rates from independent clock sources, to use the same baud rate. Allowed tolerance on this baud frequency depends on the details of how the receiver synchronizes to the leading edge of the start bit and how bit sampling is performed.

The MCU resynchronizes to bit boundaries on every high-to-low transition. In the worst case, there are no such transitions in the full 10- or 11-bit or 12-bittime character frame so any mismatch in baud rate is accumulated for the whole character time. For a Freescale UART system whose bus frequency is driven by a crystal, the allowed baud rate mismatch is about ±4.5 percent for 8-bit data format and about ±4 percent for 9-bit data format. Although baud rate modulo divider settings do not always produce baud rates that exactly match standard rates, it is normally possible to get within a few percent, which is acceptable for reliable communications.

33.4.2 Transmitter functional description

This section describes the overall block diagram for the UART transmitter, as well as specialized functions for sending break and idle characters.

The transmitter output (TxD) idle state defaults to logic high, UART_C3[TXINV] is cleared following reset. The transmitter output is inverted by setting UART_C3[TXINV]. The transmitter is enabled by setting the TE bit in UART_C2. 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 UART data register (UART_D).

The central element of the UART transmitter is the transmit shift register that is 10 or 11 or 12 bits long depending on the setting in the UART_C1[M] control bit and UART_BDH[SBNS] bit. For the remainder of this section, assume UART_C1[M] is cleared, UART_BDH[SBNS] is also 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 UART 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 (UART_S1[TDRE]) status flag is set to indicate another character may be written to the transmit data buffer at UART_D.

NOTE

Always read UART_S1 before writing to UART_D to allow data to be transmitted.

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.

Writing 0 to UART_C2[TE] does not immediately release the pin to be a general-purpose I/O pin. Any transmit activity in progress must first be completed. This includes data characters in progress, queued idle characters, and queued break characters.

33.4.2.1 Send break and queued idle

UART_C2[SBK] sends break characters originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0, 10 bit times including the start and stop bits. A longer break of 13 bit times can be enabled by setting UART_S2[BRK13]. Normally, a program would wait for UART_S1[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, write 1, and then write 0 to UART_C2[SBK]. This action queues a break character to be sent as soon as the shifter is available. If UART_C2[SBK] remains 1 when the queued break moves into the shifter, synchronized to the baud rate clock, an additional break character is queued. If the receiving device is another Freescale Semiconductor UART, the break characters are received as 0s in all eight data bits and a framing error (UART_S1[FE] = 1) occurs.

When idle-line wake-up is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for UART_S1[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the UART_C2[TE] bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while UART_C2[TE] is cleared, the UART transmitter never actually releases control of the TxD pin. If there is a possibility of the shifter finishing while UART_C2[TE] is cleared, set the general-purpose I/O controls so the pin shared with TxD is an output driving a logic 1. This ensures that the TxD line looks like a normal idle line even if the UART loses control of the port pin between writing 0 and then 1 to UART_C2[TE].

The length of the break character is affected by the UART_S2[BRK13] and UART_C1[M] as shown below.

Table 33-33. Break character length

BRK13	M	SBNS	Break character length
0	0	0	10 bit times
0	0	1	11 bit times
0	1	0	11 bit times
0	1	1	12 bit times
1	0	0	13 bit times
1	0	1	14 bit times

Table continues on the next page...

Table 33-33. Break character length (continued)

BRK13	M	SBNS	Break character length
1	1	0	14 bit times
1	1	1	15 bit times

33.4.3 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, two variations of the receiver wakeup function are explained.

The receiver input is inverted by setting UART_S2[RXINV]. The receiver is enabled by setting the UART_C2[RE] bit. Character frames consist of a start bit of logic 0, eight (or nine) data bits (lsb first), and one (or two) stop bits of logic 1. For information about 9-bit data mode, refer to 8- and 9-bit data modes. For the remainder of this discussion, assume the UART 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 (UART_S1[RDRF]) status flag is set. If UART_S1[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 UART receiver is double-buffered, the program has one full character time after UART_S1[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 (UART_S1[RDRF] = 1), it gets the data from the receive data register by reading UART_D. The UART_S1[RDRF] flag is cleared automatically by a two-step sequence normally satisfied in the course of the user's program that manages receive data. Refer to Interrupts and status flags for more details about flag clearing.

33.4.3.1 Data sampling technique

The UART receiver uses a 16× baud rate clock for sampling. The oversampling ratio is fixed at 16. The receiver starts by taking logic level samples at 16 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 16× baud rate clock divides the bit time into 16 segments labeled UART_D[RT1] through UART_D[RT16]. When a falling edge is located, three more samples are taken at UART_D[RT3], UART_D[RT5], and UART_D[RT7] 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 receive character.

The receiver then samples each bit time, including the start and stop bits, at UART_D[RT8], UART_D[RT9], and UART_D[RT10] 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. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at UART_D[RT3], UART_D[RT5], and UART_D[RT7] are 0 even if one or all of the samples taken at UART_D[RT8], UART_D[RT9], and UART_D[RT10] are 1s. 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 (UART_S1[NF]) is set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges. If an edge is detected, the sample clock is resynchronized to bit times. 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.

In the case of a framing error, the receiver is inhibited from receiving any new characters until the framing error flag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if UART_S1[FE] remains set.

33.4.3.2 Receiver wake-up operation

Receiver wake-up is a hardware mechanism that allows an UART receiver to ignore the characters in a message intended for a different UART receiver. In such a system, 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 field (UART_C2[RWU]). When UART_C2[RWU] is set, the status flags associated with the receiver, (with the exception of the idle bit, IDLE, when UART_S2[RWUID] is set), 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 UART_C2[RWU] to 0, so all receivers wake up in time to look at the first character(s) of the next message.

33.4.3.2.1 Idle-line wakeup

When wake is cleared, the receiver is configured for idle-line wakeup. In this mode, UART_C2[RWU] is cleared automatically when the receiver detects a full character time of the idle-line level. The UART_C1[M] control field selects 8-bit or 9-bit data mode and UART_BDH[SBNS] selects 1-bit or 2-bit stop bit number that determines how many bit times of idle are needed to constitute a full character time, 10 or 11 or 12 bit times because of the start and stop bits.

When UARTI_C2[RWU] is 1 and UART_S2[RWUID] is 0, the idle condition that wakes up the receiver does not set UART_S1[IDLE]. The receiver wakes up and waits for the first data character of the next message that sets UART_S1[RDRF] and generates an interrupt, if enabled. When UART_S2[RWUID] is 1, any idle condition sets UART_S1[IDLE] flag and generates an interrupt if enabled, regardless of whether UART_C2[RWU] is 0 or 1.

The idle-line type (UART_C1[ILT]) control bit selects one of two ways to detect an idle line. When UART_C1[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 UART_C1[ILT] is set, the idle bit counter does not start until after a stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

33.4.3.2.2 Address-mark wakeup

When wake is set, the receiver is configured for address-mark wakeup. In this mode, UART_C2[RWU] is cleared automatically when the receiver detects a, or two, if UART_BDH[SBNS] = 1, logic 1 in the most significant bits of a received character, eighth bit when UART_C1[M] is cleared and ninth bit when UART_C1[M] is set.

Address-mark wakeup allows messages to contain idle characters, but requires the msb be reserved for use in address frames. The one, or two, if UART_BDH[SBNS] = 1, logic 1s msb of an address frame clears the UART_C2[RWU] bit before the stop bits are received and sets the UART_S1[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.

33.4.4 Interrupts and status flags

The UART system has three separate interrupt vectors to reduce the amount of software needed to isolate the cause of the interrupt.

One interrupt vector is associated with the transmitter for UART_S1[TDRE] and UART_S1[TC] events. Another interrupt vector is associated with the receiver for RDRF, IDLE, RXEDGIF, and LBKDIF events. A third vector is used for OR, NF, FE, and PF error conditions. Each of these ten interrupt sources can be separately masked by local interrupt enable masks. The flags can be polled by software when the local masks are cleared to disable generation of hardware interrupt requests.

The UART transmitter has two status flags that can optionally generate hardware interrupt requests. Transmit data register empty (UART_S1[TDRE]) indicates when there is room in the transmit data buffer to write another transmit character to UART_D. If the transmit interrupt enable (UART_C2[TIE]) bit is set, a hardware interrupt is requested when UART_S1[TDRE] is set. Transmit complete (UART_S1[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 (UART_C2[TCIE]) bit is set, a hardware interrupt is requested when UART_S1[TC] is set. Instead of hardware interrupts, software polling may be used to monitor the UART_S1[TDRE] and UART_S1[TC] status flags if the corresponding UART_C2[TIE] or UART_C2[TCIE] local interrupt masks are cleared.

When a program detects that the receive data register is full (UART_S1[RDRF] = 1), it gets the data from the receive data register by reading UART_D. The UART_S1[RDRF] flag is cleared by reading UART_S1 while UART_S1[RDRF] is set and then reading UART_D.

When polling is used, this sequence is naturally satisfied in the normal course of the user program. If hardware interrupts are used, UART_S1 must be read in the interrupt service routine (ISR). Normally, this is done in the ISR anyway to check for receive errors, so the sequence is automatically satisfied.

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 reading UART_S1 while UART_S1[IDLE] is set and then reading UART_D. After UART_S1[IDLE] has been cleared, it cannot become set again until the receiver has received at least one new character and has set UART_S1[RDRF].

If the associated error was detected in the received character that caused UART_S1[RDRF] to be set, the error flags - noise flag (UART_S1[NF]), framing error (UART_S1[FE]), and parity error flag (UART_S1[PF]) - are set at the same time as UART_S1[RDRF]. These flags are not set in overrun cases.

If UART_S1[RDRF] was already set when a new character is ready to be transferred from the receive shifter to the receive data buffer, the overrun (UART_S1[OR]) flag is set instead of the data along with any associated NF, FE, or PF condition is lost.

At any time, an active edge on the RxD serial data input pin causes the UART_S2[RXEDGIF] flag to set. The UART_S2[RXEDGIF] flag is cleared by writing a 1 to it. This function depends on the receiver being enabled (UART_C2[RE] = 1).

33.4.5 Baud rate tolerance

A transmitting device may operate at a baud rate below or above that of the receiver.

Accumulated bit time misalignment can cause one of the three stop bit data samples (RT8, RT9, and RT10) to fall outside the actual stop bit. A noise error will occur if the RT8, RT9, and RT10 samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the RT8, RT9, and RT10 stop bit samples are a logic zero.

As the receiver samples an incoming frame, it re-synchronizes the RT clock on any valid falling edge within the frame. Resynchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

33.4.5.1 Slow data tolerance

Figure 33-31 shows how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

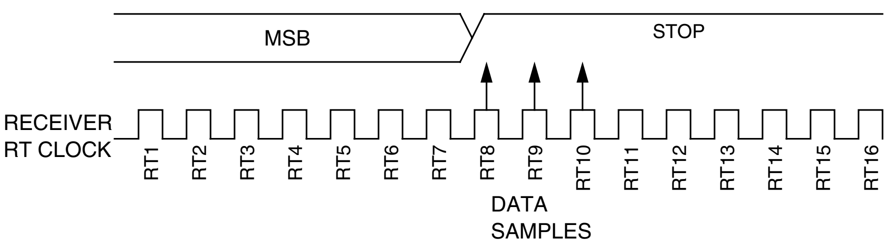


Figure 33-31. Slow data

For an 8-bit data and 1 stop bit character, data sampling of the stop bit takes the receiver 9 bit times x 16 RT cycles +10 RT cycles =154 RT cycles.

With the misaligned character shown in Figure 33-31, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 9 bit times x 16 RT cycles + 3 RT cycles = 147 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data and 1 stop bit character with no errors is:

((154 - 147) / 154) x 100 = 4.54%

For a 9-bit data or 2 stop bits character, data sampling of the stop bit takes the receiver 10 bit times x 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in Figure 33-31, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 10 bit times x 16 RT cycles + 3 RT cycles = 163 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit or 2 stop bits character with no errors is:

((170 - 163) / 170) X 100 = 4.12%

For a 9-bit data and 2 stop bit character, data sampling of the stop bit takes the receiver 11 bit times x 16 RT cycles + 10 RT cycles = 186 RT cycles.

With the misaligned character shown in Figure 33-31, the receiver counts 186 RT cycles at the point when the count of the transmitting device is 11 bit times x 16 RT cycles + 3 RT cycles = 179 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit and 2 stop bits character with no errors is: ((186 - 179) / 186) X 100 = 3.76%

33.4.5.2 Fast data tolerance

Figure 33-32 shows how much a fast received frame can be misaligned. The fast stop bit ends at RT10 instead of RT16 but is still sampled at RT8, RT9, and RT10.

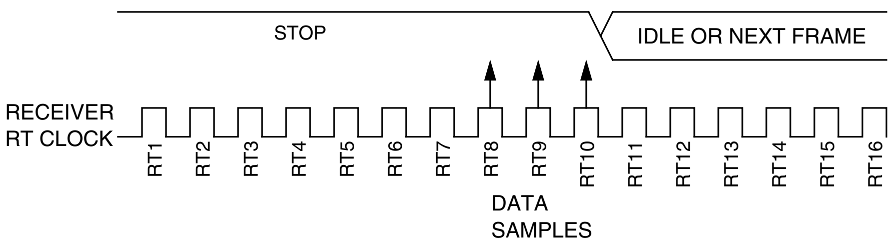


Figure 33-32. Fast data

For an 8-bit data and 1 stop bit character, data sampling of the stop bit takes the receiver 9 bit times x 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in Figure 33-32, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 10 bit times x 16 RT cycles = 160 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit and 1 stop bit character with no errors is:

((154 - 160) / 154) x 100 = 3.90%

For a 9-bit data or 2 stop bits character, data sampling of the stop bit takes the receiver 10 bit times x 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 11 bit times x 16 RT cycles = 176 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit or 2 stop bits character with no errors is:

((170 - 176) / 170) x 100 = 3.53%

For a 9-bit data and 2 stop bits character, data sampling of the stop bit takes the receiver 11 bit times x 16 RT cycles + 10 RT cycles = 186 RT cycles.

With the misaligned character shown in, the receiver counts 186 RT cycles at the point when the count of the transmitting device is 12 bit times x 16 RT cycles = 192 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit and 2 stop bits character with no errors is:

((186 - 192) / 186) x 100 = 3.23%

33.4.6 DMA Operation

In the transmitter, flags TDRE and TC can be configured to assert a DMA transfer request. In the receiver, flags RDRF, IDLE and LBKDIF can be configured to assert a DMA transfer request.

The table found here shows the configuration bit settings required to configure each flag for DMA operation.

Table 33-34. DMA configuration

Flag	Request enable bit	DMA select bit
TDRE	TIE = 1	TDMAS = 1
TC	TCIE = 1	TCDMAS = 1
RDRF	RIE = 1	RDMAS = 1
IDLE	ILIE = 1	ILDMAS = 1
LBKDIF	LBKDIE = 1	LBKDDMAS = 1

When a flag is configured for a DMA request, its associated DMA request is asserted when the flag is set. When the RDRF or IDLE flag is configured as a DMA request, the clearing mechanism of reading UART_S1 followed by reading UART_D does not clear the associated flag.The DMA request remains asserted until an indication is received that the DMA transactions are done. When this indication is received, the flag bit and the associated DMA request are cleared. If the DMA operation failed to remove the situation that caused the DMA request another request will be issued.

33.4.7 Additional UART functions

The following sections describe additional UART functions.

33.4.7.1 8- and 9-bit data modes

The UART system, transmitter and receiver, can be configured to operate in 9-bit data mode by setting UART_C1[M]. In 9-bit mode, there is a ninth data bit to the left of the most significant bit of the UART data register. For the transmit data buffer, this bit is stored in T8 in UART_C3. For the receiver, the ninth bit is held in UART_C3[R8].

For coherent writes to the transmit data buffer, write to UART_C3[T8] before writing to UART_D.

If the bit value to be transmitted as the ninth bit of a new character is the same as for the previous character, it is not necessary to write to UART_C3[T8] again. When data is transferred from the transmit data buffer to the transmit shifter, the value in UART_C3[T8] is copied at the same time data is transferred from UART_D 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 wake-up so the ninth data bit can serve as the wakeup bit. In custom protocols, the ninth bit can also serve as a software-controlled marker.

33.4.7.2 Stop mode operation

During all stop modes, clocks to the UART module are halted.

No UART module registers are affected in Stop3 mode.

The receive input active edge detect circuit remains active in Stop3 mode. An active edge on the receive input brings the CPU out of stop and VLPS mode if the interrupt is not masked (UART_BDH[RXEDGIE] = 1).

Because the clocks are halted, the UART module resumes operation upon exit from stop, only in stop and VLPS mode. Software must ensure stop mode is not entered while there is a character (including preamble, break and normal data) being transmitted out of or received into the UART module, that means UART_S1[TC] =1, UART_S1[TDRE] = 1, and UART_S2[RAF] = 0 must all meet before entering stop mode.

33.4.7.3 Loop mode

When UART_C1[LOOPS] is set, the UART_C1[RSRC] bit in the same register chooses between loop mode (UART_C1[RSRC] = 0) or single-wire mode (UART_C1[RSRC] = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the internal loop back connection from the transmitter to the receiver causes the receiver to receive characters that are sent out by the transmitter.

33.4.7.4 Single-wire operation

When UART_C1[LOOPS] is set, UART_C1[RSRC] chooses between loop mode (UART_C1[RSRC] = 0) or single-wire mode (UART_C1[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 and reverts to a general-purpose port I/O pin.

In single-wire mode, the UART_C3[TXDIR] bit controls the direction of serial data on the TxD pin. When UART_C3[TXDIR] is cleared, the TxD pin is an input to the UART receiver and the transmitter is temporarily disconnected from the TxD pin so an external device can send serial data to the receiver. When UART_C3[TXDIR] is set, the TxD pin is an output driven by the transmitter. In single-wire mode, the transmitter output is internally connected to the receiver input and the RxD pin is not used by the UART, so it reverts to a general-purpose port I/O pin.

Chapter 34
General-Purpose Input/Output (GPIO)

34.1 Introduction

NOTE

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

The general-purpose input and output (GPIO) module is accessible via the peripheral bus and also communicates to the processor core via a zero wait state interface (IOPORT) for maximum pin performance. The GPIO registers support 8-bit, 16-bit or 32-bit accesses.

The GPIO data direction and output data registers control the direction and output data of each pin when the pin is configured for the GPIO function. The GPIO input data register displays the logic value on each pin when the pin is configured for any digital function, provided the corresponding Port Control and Interrupt module for that pin is enabled.

Efficient bit manipulation of the general-purpose outputs is supported through the addition of set, clear, and toggle write-only registers for each port output data register.

34.1.1 Features

- Features of the GPIO module include:
  - Port Data Input register visible in all digital pin-multiplexing modes
  - Port Data Input register with corresponding set/clear/toggle registers
  - Port Data Direction register
  - Zero wait state access to GPIO registers through IOPORT

NOTE

The GPIO module is clocked by system clock.

34.1.2 Modes of operation

The following table depicts different modes of operation and the behavior of the GPIO module in these modes.

Table 34-1. 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.

34.1.3 GPIO signal descriptions

Table 34-2. 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.

34.1.3.1 Detailed signal description

Table 34-3. GPIO interface-detailed signal descriptions

Signal	I/O	Description	
PORTA31–PORTA0 PORTB31–PORTB0 PORTC31–PORTC0 PORTD31–PORTD0 PORTE31–PORTE0	I/O	General-purpose input/output	
		State meaning	Asserted: The pin is logic 1. Deasserted: The pin is logic 0.
		Timing	Assertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock. 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.

34.2 Memory map and register definition

Any read or write access to the GPIO memory space that is outside the valid memory map results in a bus error.

NOTE

For simplicity, each GPIO port's registers appear with the same width of 32 bits, corresponding to 32 pins. The actual number of pins per port (and therefore the number of usable control bits per port register) is chip-specific. Refer to the Chip Configuration chapter to see the exact control bits for the non-identical port instance.

GPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F000	Port Data Output Register (GPIOA_PDOR)	32	R/W	0000_0000h	34.2.1/667
400F_F004	Port Set Output Register (GPIOA_PSOR)	32	W (always reads 0)	0000_0000h	34.2.2/668

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F008	Port Clear Output Register (GPIOA_PCOR)	32	W (always reads 0)	0000_0000h	34.2.3/668
400F_F00C	Port Toggle Output Register (GPIOA_PTOR)	32	W (always reads 0)	0000_0000h	34.2.4/669
400F_F010	Port Data Input Register (GPIOA_PDIR)	32	R	0000_0000h	34.2.5/669
400F_F014	Port Data Direction Register (GPIOA_PDDR)	32	R/W	0000_0000h	34.2.6/670
400F_F040	Port Data Output Register (GPIOB_PDOR)	32	R/W	0000_0000h	34.2.1/667
400F_F044	Port Set Output Register (GPIOB_PSOR)	32	W (always reads 0)	0000_0000h	34.2.2/668
400F_F048	Port Clear Output Register (GPIOB_PCOR)	32	W (always reads 0)	0000_0000h	34.2.3/668
400F_F04C	Port Toggle Output Register (GPIOB_PTOR)	32	W (always reads 0)	0000_0000h	34.2.4/669
400F_F050	Port Data Input Register (GPIOB_PDIR)	32	R	0000_0000h	34.2.5/669
400F_F054	Port Data Direction Register (GPIOB_PDDR)	32	R/W	0000_0000h	34.2.6/670
400F_F080	Port Data Output Register (GPIOC_PDOR)	32	R/W	0000_0000h	34.2.1/667
400F_F084	Port Set Output Register (GPIOC_PSOR)	32	W (always reads 0)	0000_0000h	34.2.2/668
400F_F088	Port Clear Output Register (GPIOC_PCOR)	32	W (always reads 0)	0000_0000h	34.2.3/668
400F_F08C	Port Toggle Output Register (GPIOC_PTOR)	32	W (always reads 0)	0000_0000h	34.2.4/669
400F_F090	Port Data Input Register (GPIOC_PDIR)	32	R	0000_0000h	34.2.5/669
400F_F094	Port Data Direction Register (GPIOC_PDDR)	32	R/W	0000_0000h	34.2.6/670
400F_F0C0	Port Data Output Register (GPIOD_PDOR)	32	R/W	0000_0000h	34.2.1/667
400F_F0C4	Port Set Output Register (GPIOD_PSOR)	32	W (always reads 0)	0000_0000h	34.2.2/668
400F_F0C8	Port Clear Output Register (GPIOD_PCOR)	32	W (always reads 0)	0000_0000h	34.2.3/668
400F_F0CC	Port Toggle Output Register (GPIOD_PTOR)	32	W (always reads 0)	0000_0000h	34.2.4/669
400F_F0D0	Port Data Input Register (GPIOD_PDIR)	32	R	0000_0000h	34.2.5/669
400F_F0D4	Port Data Direction Register (GPIOD_PDDR)	32	R/W	0000_0000h	34.2.6/670

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F100	Port Data Output Register (GPIOE_PDOR)	32	R/W	0000_0000h	34.2.1/667
400F_F104	Port Set Output Register (GPIOE_PSOR)	32	W (always reads 0)	0000_0000h	34.2.2/668
400F_F108	Port Clear Output Register (GPIOE_PCOR)	32	W (always reads 0)	0000_0000h	34.2.3/668
400F_F10C	Port Toggle Output Register (GPIOE_PTOR)	32	W (always reads 0)	0000_0000h	34.2.4/669
400F_F110	Port Data Input Register (GPIOE_PDIR)	32	R	0000_0000h	34.2.5/669
400F_F114	Port Data Direction Register (GPIOE_PDDR)	32	R/W	0000_0000h	34.2.6/670

34.2.1 Port Data Output Register (GPIOx_PDOR)

This register configures the logic levels that are driven on each general-purpose output pins.

NOTE

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

Address: Base address + 0h offset

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

GPIOx_PDOR field descriptions

Field	Description
31–0 PDO	Port Data Output Register bits for unbonded pins return a undefined value when read. 0 Logic level 0 is driven on pin, provided pin is configured for general-purpose output. 1 Logic level 1 is driven on pin, provided pin is configured for general-purpose output.

34.2.2 Port Set Output Register (GPIOx_PSOR)

This register configures whether to set the fields of the PDOR.

Address: Base address + 4h offset

Bit	31–16	15–0
R	0	
W	PTSO	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

GPIOx_PSOR field descriptions

Field	Description
31–0 PTSO	Port Set Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to logic 1.

34.2.3 Port Clear Output Register (GPIOx_PCOR)

This register configures whether to clear the fields of PDOR.

Address: Base address + 8h offset

Bit	31–16	15–0
R	0	
W	PTCO	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

GPIOx_PCOR field descriptions

Field	Description
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: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is cleared to logic 0.

34.2.4 Port Toggle Output Register (GPIOx_PTOR)

Address: Base address + Ch offset

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

GPIOx_PTOR field descriptions

Field	Description
31–0 PTTO	Port Toggle Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to the inverse of its existing logic state.

34.2.5 Port Data Input Register (GPIOx_PDIR)

NOTE

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

Address: Base address + 10h offset

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

GPIOx_PDIR field descriptions

Field	Description
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. 0 Pin logic level is logic 0, or is not configured for use by digital function. 1 Pin logic level is logic 1.

34.2.6 Port Data Direction Register (GPIOx_PDDR)

The PDDR configures the individual port pins for input or output.

Address: Base address + 14h offset

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

GPIOx_PDDR field descriptions

Field	Description
31–0 PDD	Port Data Direction Configures individual port pins for input or output. 0 Pin is configured as general-purpose input, for the GPIO function. 1 Pin is configured as general-purpose output, for the GPIO function.

34.3 FGPIO memory map and register definition

The GPIO registers are also aliased to the IOPORT interface on the Cortex-M0+ from address 0xF800_0000.

Accesses via the IOPORT interface occur in parallel with any instruction fetches and will therefore complete in a single cycle. This aliased Fast GPIO memory map is called FGPIO.

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

NOTE

For simplicity, each FGPIO port's registers appear with the same width of 32 bits, corresponding to 32 pins. The actual number of pins per port (and therefore the number of usable control bits per port register) is chip-specific. Refer to the Chip Configuration chapter to see the exact control bits for the non-identical port instance.

FGPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F800_0000	Port Data Output Register (FGPIOA_PDOR)	32	R/W	0000_0000h	34.3.1/672
F800_0004	Port Set Output Register (FGPIOA_PSOR)	32	W (always reads 0)	0000_0000h	34.3.2/673
F800_0008	Port Clear Output Register (FGPIOA_PCOR)	32	W (always reads 0)	0000_0000h	34.3.3/673
F800_000C	Port Toggle Output Register (FGPIOA_PTOR)	32	W (always reads 0)	0000_0000h	34.3.4/674
F800_0010	Port Data Input Register (FGPIOA_PDIR)	32	R	0000_0000h	34.3.5/674
F800_0014	Port Data Direction Register (FGPIOA_PDDR)	32	R/W	0000_0000h	34.3.6/675
F800_0040	Port Data Output Register (FGPIOB_PDOR)	32	R/W	0000_0000h	34.3.1/672
F800_0044	Port Set Output Register (FGPIOB_PSOR)	32	W (always reads 0)	0000_0000h	34.3.2/673
F800_0048	Port Clear Output Register (FGPIOB_PCOR)	32	W (always reads 0)	0000_0000h	34.3.3/673
F800_004C	Port Toggle Output Register (FGPIOB_PTOR)	32	W (always reads 0)	0000_0000h	34.3.4/674
F800_0050	Port Data Input Register (FGPIOB_PDIR)	32	R	0000_0000h	34.3.5/674
F800_0054	Port Data Direction Register (FGPIOB_PDDR)	32	R/W	0000_0000h	34.3.6/675
F800_0080	Port Data Output Register (FGPIOC_PDOR)	32	R/W	0000_0000h	34.3.1/672
F800_0084	Port Set Output Register (FGPIOC_PSOR)	32	W (always reads 0)	0000_0000h	34.3.2/673
F800_0088	Port Clear Output Register (FGPIOC_PCOR)	32	W (always reads 0)	0000_0000h	34.3.3/673
F800_008C	Port Toggle Output Register (FGPIOC_PTOR)	32	W (always reads 0)	0000_0000h	34.3.4/674
F800_0090	Port Data Input Register (FGPIOC_PDIR)	32	R	0000_0000h	34.3.5/674
F800_0094	Port Data Direction Register (FGPIOC_PDDR)	32	R/W	0000_0000h	34.3.6/675
F800_00C0	Port Data Output Register (FGPIOD_PDOR)	32	R/W	0000_0000h	34.3.1/672
F800_00C4	Port Set Output Register (FGPIOD_PSOR)	32	W (always reads 0)	0000_0000h	34.3.2/673
F800_00C8	Port Clear Output Register (FGPIOD_PCOR)	32	W (always reads 0)	0000_0000h	34.3.3/673

Table continues on the next page...

FGPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F800_00CC	Port Toggle Output Register (FGPIOD_PTOR)	32	W (always reads 0)	0000_0000h	34.3.4/674
F800_00D0	Port Data Input Register (FGPIOD_PDIR)	32	R	0000_0000h	34.3.5/674
F800_00D4	Port Data Direction Register (FGPIOD_PDDR)	32	R/W	0000_0000h	34.3.6/675
F800_0100	Port Data Output Register (FGPIOE_PDOR)	32	R/W	0000_0000h	34.3.1/672
F800_0104	Port Set Output Register (FGPIOE_PSOR)	32	W (always reads 0)	0000_0000h	34.3.2/673
F800_0108	Port Clear Output Register (FGPIOE_PCOR)	32	W (always reads 0)	0000_0000h	34.3.3/673
F800_010C	Port Toggle Output Register (FGPIOE_PTOR)	32	W (always reads 0)	0000_0000h	34.3.4/674
F800_0110	Port Data Input Register (FGPIOE_PDIR)	32	R	0000_0000h	34.3.5/674
F800_0114	Port Data Direction Register (FGPIOE_PDDR)	32	R/W	0000_0000h	34.3.6/675

34.3.1 Port Data Output Register (FGPIOx_PDOR)

This register configures the logic levels that are driven on each general-purpose output pins.

Address: Base address + 0h offset

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

FGPIOx_PDOR field descriptions

Field	Description
31–0 PDO	Port Data Output Unimplemented pins for a particular device read as zero. 0 Logic level 0 is driven on pin, provided pin is configured for general-purpose output. 1 Logic level 1 is driven on pin, provided pin is configured for general-purpose output.

34.3.2 Port Set Output Register (FGPIOx_PSOR)

This register configures whether to set the fields of the PDOR.

Address: Base address + 4h offset

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

FGPIOx_PSOR field descriptions

Field	Description
31–0 PTSO	Port Set Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to logic 1.

34.3.3 Port Clear Output Register (FGPIOx_PCOR)

This register configures whether to clear the fields of PDOR.

Address: Base address + 8h offset

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

FGPIOx_PCOR field descriptions

Field	Description
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: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is cleared to logic 0.

34.3.4 Port Toggle Output Register (FGPIOx_PTOR)

Address: Base address + Ch offset

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

FGPIOx_PTOR field descriptions

Field	Description
31–0 PTTO	Port Toggle Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to the inverse of its existing logic state.

34.3.5 Port Data Input Register (FGPIOx_PDIR)

Address: Base address + 10h offset

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

FGPIOx_PDIR field descriptions

Field	Description
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. 0 Pin logic level is logic 0, or is not configured for use by digital function. 1 Pin logic level is logic 1.

34.3.6 Port Data Direction Register (FGPIOx_PDDR)

The PDDR configures the individual port pins for input or output.

Address: Base address + 14h offset

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

FGPIOx_PDDR field descriptions

Field	Description
31–0 PDD	Port Data Direction Configures individual port pins for input or output. 0 Pin is configured as general-purpose input, for the GPIO function. 1 Pin is configured as general-purpose output, for the GPIO function.

34.4 Functional description

34.4.1 General-purpose input

The logic state of each pin is available via the Port Data Input registers, provided the pin is configured for a digital function and the corresponding Port Control and Interrupt module is enabled.

34.4.2 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.

34.4.3 IOPORT

The GPIO registers are also aliased to the IOPORT interface on the Cortex-M0+ from address 0xF800_0000. Accesses via the IOPORT interface occur in parallel with any instruction fetches and will therefore complete in a single cycle. If the DMA attempts to access the GPIO registers on the same cycle as an IOPORT access, then the DMA access will stall until any IOPORT accesses have completed.

During Compute Operation, the GPIO registers remain accessible via the IOPORT interface only. Since the clocks to the Port Control and Interrupt modules are disabled during Compute Operation, the Pin Data Input Registers do not update with the current state of the pins.

Chapter 35 Touch Sensing Input (TSI)

35.1 Introduction

The touch sensing input (TSI) module provides capacitive touch sensing detection with high sensitivity and enhanced robustness.

Each TSI pin implements the capacitive measurement by a current source scan, charging and discharging the electrode, once or several times. A reference oscillator ticks the scan time and stores the result in a 16-bit register when the scan completes. Meanwhile, an interrupt request is submitted to CPU for post-processing if TSI interrupt is enabled and DMA function is not selected.The TSI module can be periodically triggered to work in low power mode with ultra-low current adder and wake CPU at the end of scan or the conversion result is out of the range specified by TSI threshold. It provides a solid capacitive measurement module to the implementation of touch keyboard, rotaries and sliders.

35.1.1 Features

TSI features includes:

- Support up to 16 external electrodes
- Automatic detection of electrode capacitance across all operational power modes
- Internal reference oscillator for high-accuracy measurement
- Configurable software or hardware scan trigger
- Fully support Freescale touch sensing software (TSS) library, see www.freescale.com/touchsensing.
- Capability to wake MCU from low power modes
- Compensate for temperature and supply voltage variations
- High sensitivity change with 16-bit resolution register
- Configurable up to 4096 scan times.

- Support DMA data transfer
- For electrode design recommendations refer to AN3863: Designing Touch Sensing Electrodes

35.1.2 Modes of operation

This module supports the following operation modes.

Table 35-1. Operating modes

Mode	Description
Stop and low power stop	TSI module is fully functional in all of the stop modes as long as TSI_GENCS[STPE] is set. The channel specified by TSI_DATA[TSICH] will be scanned upon the trigger. After scan finishes, either end-of-scan or out-of-range interrupt can be selected to bring MCU out of low power modes.
Wait	TSI module is fully functional in this mode. When a scan completes, TSI submits an interrupt request to CPU if the interrupt is enabled.
Run	TSI module is fully functional in this mode. When a scan completes, TSI submits an interrupt request to CPU if the interrupt is enabled.

35.1.3 Block diagram

The following figure is a block diagram of the TSI module.

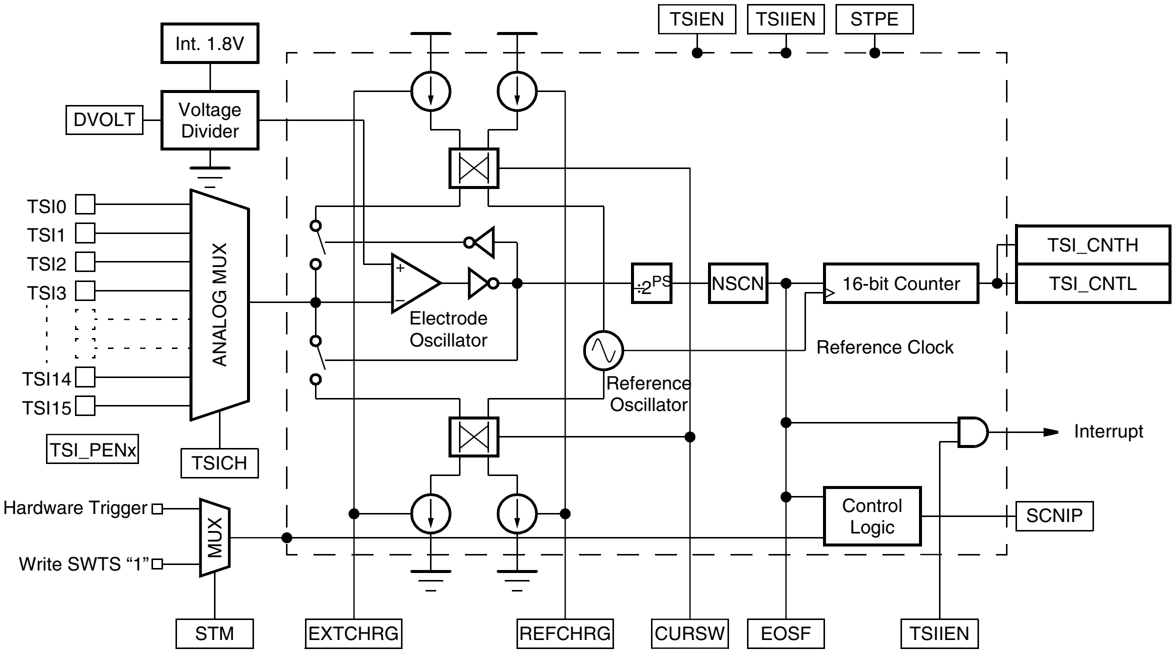


Figure 35-1. TSI module block diagram

35.2 External signal description

The TSI module contains up to 16 external pins for touch sensing. The table found here describes each of the TSI external pins.

Table 35-2. TSI signal description

Name	Port	Direction	Function	Reset state
TSI[15:0]	TSI	I/O	TSI capacitive pins. Switches driver that connects directly to the electrode pins TSI[15:0] can operate as GPIO pins.	I/O

35.2.1 TSI[15:0]

When TSI functionality is enabled , the TSI analog portion uses the corresponding channel to connect external on-board touch capacitors. The PCB connection between the pin and the touch pad must be kept as short as possible to reduce distribution capacity on board.

35.3 Register definition

This section describes the memory map and control/status registers for the TSI module.

TSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_5000	TSI General Control and Status Register (TSI0_GENCS)	32	R/W	0000_0000h	35.3.1/680
4004_5004	TSI DATA Register (TSI0_DATA)	32	R/W	0000_0000h	35.3.2/684
4004_5008	TSI Threshold Register (TSI0_TSHD)	32	R/W	0000_0000h	35.3.3/686

35.3.1 TSI General Control and Status Register (TSIx_GENCS)

This control register provides various control and configuration information for the TSI module.

NOTE

When TSI is working, the configuration bits (GENCS[TSIEN], GENCS[TSIIEN], and GENCS[STM]) must not be changed. The EOSF flag is kept until the software acknowledge it.

Address: 4004_5000h base + 0h offset = 4004_5000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OUTRGF	0		ESOR	MODE				REFCHRG			DVOLT		EXTCHRG		
W	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PS			NSCN					TSIEN	TSIIEN	STPE	STM	SCNIP	EOSF	CURSW	0
W														w1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSIx_GENCS field descriptions

Field	Description
31 OUTRGF	Out of Range Flag. This flag is set if the result register of the enabled electrode is out of the range defined by the TSI_THRESHOLD register. This flag is set only when TSI is configured in non-noise detection mode. It can be read once the CPU wakes. Write "1" , when this flag is set, to clear it.
30–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 ESOR	End-of-scan or Out-of-Range Interrupt Selection This bit is used to select out-of-range or end-of-scan event to generate an interrupt. 0 Out-of-range interrupt is allowed. 1 End-of-scan interrupt is allowed.
27–24 MODE	TSI analog modes setup and status bits. Set up TSI analog modes, especially, setting MODE[3:2] to not 2'b00 will configure TSI to noise detection modes. MODE[1:0] take no effect on TSI operation mode and should always write to 2'b00 for setting up. When reading this field will return the analog status. Refer to chapter "Noise detection mode" for details. 0000 Set TSI in capacitive sensing(non-noise detection) mode. 0100 Set TSI analog to work in single threshold noise detection mode and the frequency limitation circuit is disabled. 1000 Set TSI analog to work in single threshold noise detection mode and the frequency limitation circuit is enabled to work in higher frequencies operations. 1100 Set TSI analog to work in automatic noise detection mode.
23–21 REFCHRG	REFCHRG These bits indicate the reference oscillator charge and discharge current value. 000 500 nA. 001 1 μA. 010 2 μA. 011 4 μA. 100 8 μA.

Table continues on the next page...

TSIx_GENCS field descriptions (continued)

Field	Description
	101 16 μA. 110 32 μA. 111 64 μA.
20–19 DVOLT	DVOLT These bits indicate the oscillator's voltage rails as below. 00 DV = 1.026 V; V_P = 1.328 V; V_m = 0.302 V. 01 DV = 0.592 V; V_P = 1.111 V; V_m = 0.519 V. 10 DV = 0.342 V; V_P = 0.986 V; V_m = 0.644 V. 11 DV = 0.197 V; V_P = 0.914 V; V_m = 0.716 V.
18–16 EXTCHRG	EXTCHRG These bits indicate the electrode oscillator charge and discharge current value. 000 500 nA. 001 1 μA. 010 2 μA. 011 4 μA. 100 8 μA. 101 16 μA. 110 32 μA. 111 64 μA.
15–13 PS	PS These bits indicate the prescaler of the output of electrode oscillator. 000 Electrode Oscillator Frequency divided by 1 001 Electrode Oscillator Frequency divided by 2 010 Electrode Oscillator Frequency divided by 4 011 Electrode Oscillator Frequency divided by 8 100 Electrode Oscillator Frequency divided by 16 101 Electrode Oscillator Frequency divided by 32 110 Electrode Oscillator Frequency divided by 64 111 Electrode Oscillator Frequency divided by 128
12–8 NSCN	NSCN These bits indicate the scan number for each electrode. The scan number is equal to NSCN + 1, which allows the scan time ranges from 1 to 32. By default, NSCN is configured as 0, which asserts the TSI scans once on the selected eletrode channel. 00000 Once per electrode 00001 Twice per electrode 00010 3 times per electrode 00011 4 times per electrode 00100 5 times per electrode 00101 6 times per electrode 00110 7 times per electrode 00111 8 times per electrode

Table continues on the next page...

TSIx_GENCS field descriptions (continued)

Field	Description
	01000 9 times per electrode 01001 10 times per electrode 01010 11 times per electrode 01011 12 times per electrode 01100 13 times per electrode 01101 14 times per electrode 01110 15 times per electrode 01111 16 times per electrode 10000 17 times per electrode 10001 18 times per electrode 10010 19 times per electrode 10011 20 times per electrode 10100 21 times per electrode 10101 22 times per electrode 10110 23 times per electrode 10111 24 times per electrode 11000 25 times per electrode 11001 26 times per electrode 11010 27 times per electrode 11011 28 times per electrode 11100 29 times per electrode 11101 30 times per electrode 11110 31 times per electrode 11111 32 times per electrode
7 TSIEN	Touch Sensing Input Module Enable This bit enables TSI module. 0 TSI module disabled. 1 TSI module enabled.
6 TSIIEN	Touch Sensing Input Interrupt Enable This bit enables TSI module interrupt request to CPU when the scan completes. The interrupt will wake MCU from low power mode if this interrupt is enabled. 0 TSI interrupt is disabled. 1 TSI interrupt is enabled.
5 STPE	TSI STOP Enable This bit enables TSI module function in low power modes (stop, VLPS, LLS and VLLS{3,2,1}). 0 TSI is disabled when MCU goes into low power mode. 1 Allows TSI to continue running in all low power modes.
4 STM	Scan Trigger Mode This bit specifies the trigger mode. User is allowed to change this bit when TSI is not working in progress. 0 Software trigger scan. 1 Hardware trigger scan.

Table continues on the next page...

TSIx_GENCS field descriptions (continued)

Field	Description
3 SCNIP	Scan In Progress Status This read-only bit indicates if scan is in progress. This bit will get asserted after the analog bias circuit is stable after a trigger and it changes automatically by the TSI. 0 No scan in progress. 1 Scan in progress.
2 EOSF	End of Scan Flag This flag is set when all active electrodes are finished scanning after a scan trigger. Write "1" , when this flag is set, to clear it. 0 Scan not complete. 1 Scan complete.
1 CURSW	CURSW This bit specifies if the current sources of electrode oscillator and reference oscillator are swapped. 0 The current source pair are not swapped. 1 The current source pair are swapped.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

35.3.2 TSI DATA Register (TSIx_DATA)

Address: 4004_5000h base + 4h offset = 4004_5004h

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

TSIx_DATA field descriptions

Field	Description
31–28 TSICH	TSICH These bits specify current channel to be measured. In hardware trigger mode (TSI_GENCS[STM] = 1), the scan will not start until the hardware trigger occurs. In software trigger mode (TSI_GENCS[STM] = 0), the scan starts immediately when TSI_DATA[SWTS] bit is written by 1. 0000 Channel 0. 0001 Channel 1. 0010 Channel 2. 0011 Channel 3. 0100 Channel 4. 0101 Channel 5. 0110 Channel 6. 0111 Channel 7. 1000 Channel 8. 1001 Channel 9. 1010 Channel 10. 1011 Channel 11. 1100 Channel 12. 1101 Channel 13. 1110 Channel 14. 1111 Channel 15.
27–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 DMAEN	DMA Transfer Enabled This bit is used together with the TSI interrupt enable bits(TSIIE, ESOR) to generate a DMA transfer request instead of an interrupt. 0 Interrupt is selected when the interrupt enable bit is set and the corresponding TSI events assert. 1 DMA transfer request is selected when the interrupt enable bit is set and the corresponding TSI events assert.
22 SWTS	Software Trigger Start This write-only bit is a software start trigger. When STM bit is clear, write "1" to this bit will start a scan. The electrode channel to be scanned is determinated by TSI_DATA[TSICH] bits. 0 No effect. 1 Start a scan to determine which channel is specified by TSI_DATA[TSICH].
21–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 TSICNT	TSI Conversion Counter Value These read-only bits record the accumulated scan counter value ticked by the reference oscillator.

35.3.3 TSI Threshold Register (TSIx_TSHD)

Address: 4004_5000h base + 8h offset = 4004_5008h

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

TSIx_TSHD field descriptions

Field	Description
31–16 THRESH	TSI Wakeup Channel High-threshold This half-word specifies the high threshold of the wakeup channel.
15–0 THRESL	TSI Wakeup Channel Low-threshold This half-word specifies the low threshold of the wakeup channel.

35.4 Functional description

35.4.1 Capacitance measurement

The electrode pin capacitance measurement uses a dual oscillator approach. The frequency of the TSI electrode oscillator depends on the external electrode capacitance and the TSI module configuration. After going to a configurable prescaler, the TSI electrode oscillator signal goes to the input of the module counter. The time for the module counter to reach its module value is measured using the TSI reference oscillator. The measured electrode capacitance is directly proportional to the time.

35.4.1.1 TSI electrode oscillator

The TSI electrode oscillator circuit is illustrated in the following figure. A configurable constant current source is used to charge and discharge the external electrode capacitance. A buffer hysteresis defines the oscillator delta voltage. The delta voltage defines the margin of high and low voltage which are the reference input of the comparator in different time.

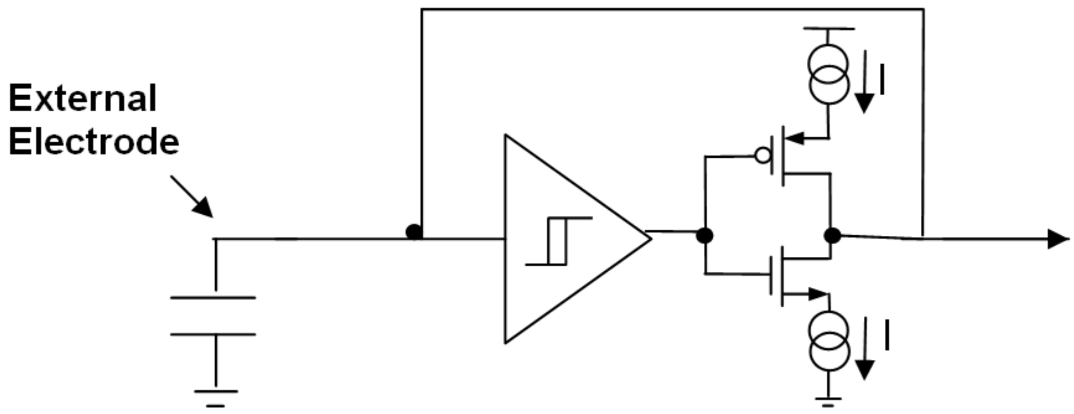


Figure 35-8. TSI electrode oscillator circuit

The current source applied to the pad capacitance is controlled by the GENCS[EXTCHRG]. The hysteresis delta voltage is defined in the module electrical specifications present in the device Data Sheet. The figure below shows the voltage amplitude waveform of the electrode capacitance charging and discharging with a programmable current.

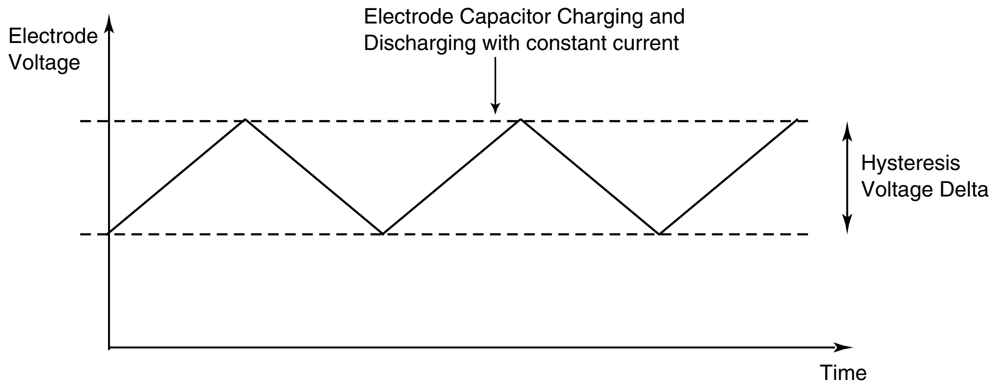


Figure 35-9. TSI electrode oscillator chart

The oscillator frequency is give by the following equation

$$F_{elec} = \frac{I}{2 * C_{elec} * \Delta V}$$

Figure 35-10. TSI electrode oscillator frequency

Where:

I: constant current

C_{elec}: electrode capacitance

ΔV: Hysteresis delta voltage

So by this equation, for example, an electrode with C_{elec}= 20 pF, with a current source of I = 16 µA and ΔV = 600 mV have the following oscillation frequency:

$$F_{elec} = \frac{16\ \mu A}{2 * 20pF * 600mV} = 0.67MHz$$

Figure 35-11. TSI electrode oscillator frequency

The current source is used to accommodate the TSI electrode oscillator frequency with different electrode capacitance sizes.

35.4.1.2 Electrode oscillator and counter module control

The TSI oscillator frequency signal goes through a prescaler defined by the GENCS[PS] and then enters in a modulus counter. GENCS[NSCN] defines the maximum count value of the modulus counter.

The pin capacitance sampling time is given by the time the module counter takes to go from 0 to its maximum value, defined by NSCN. The electrode sample time is expressed by the following equation:

$$T_{cap\_samp} = \frac{PS * NSCN}{F_{elec}}$$

Using Equation 1

$$T_{cap\_samp} = \frac{2 * PS * NSCN * C_{elec} * \Delta V}{I}$$

Figure 35-12. Electrode sampling time

Where:

PS: prescaler value

NSCN: module counter maximum value

I: constant current

C_{elec}: electrode capacitance

ΔV: Hysteresis delta voltage

By this equation we have that an electrode with C = 20 pF, with a current source of I = 16 µA and ΔV = 600 mV, PS = 2 and NSCN = 16 have the following sampling time:

$$T_{cap\_samp} = \frac{2*2*16*20pF*600mV}{16\mu A} = 48\mu s$$

35.4.1.3 TSI reference oscillator

The TSI reference oscillator has the same topology of the TSI electrode oscillator. The TSI reference oscillator instead of using an external capacitor for the electrode oscillator has an internal reference capacitor.

The TSI reference oscillator has an independent programmable current source controlled by GENCS[REFCHRG].

The reference oscillator frequency is given by the following equation:

$$F_{ref\_osc} = \frac{I_{ref}}{2 * C_{ref} * \Delta V}$$

Figure 35-13. TSI reference oscillator frequency

Where:

C_{ref}: Internal reference capacitor

I_{ref}: Reference oscillator current source

ΔV : Hysteresis delta voltage

Considering C_{ref} = 1.0 pF, I_{ref} = 12 µA and ΔV = 600 mV, follows

$$F_{ref\_osc} = \frac{12\mu A}{2 * 1.0pF * 600mV} = 10.0MHz$$

35.4.2 TSI measurement result

The capacitance measurement result is defined by the number of TSI reference oscillator periods during the sample time and is stored in the TSICHnCNT register.

$TSICHnCNT = T_{cap\_samp} * F_{ref\_osc}$

Using Equation 2 and Equation 1 follows:

$$TSICHnCNT = \frac{I_{ref} * PS * NSCN}{C_{ref} * I_{elec}} * C_{elec}$$

Figure 35-14. Capacitance result value

In the example where F_{ref_osc} = 10.0MHz and T_{cap_samp} = 48 µs, TSICHnCNT = 480

35.4.3 Enable TSI module

The TSI module can be fully functional in run, wait and low power modes. The TSI_GENCS[TSIEN] bit must be set to enable the TSI module in run and wait mode. When TSI_GENCS[STPE] bit is set, it allows the TSI module to work in low power mode.

35.4.4 Software and hardware trigger

The TSI module allows a software or hardware trigger to start a scan. When a software trigger is applied (TSI_GENCS[STM] bit clear), the TSI_GENCS[SWTS] bit must be written "1" to start the scan electrode channel that is identified by TSI_DATA[TSICH]. When a hardware trigger is applied (TSI_GENCS[STM] bit set), the TSI will not start scanning until the hardware trigger arrives. The hardware trigger is different depending on the MCU configuration. Generally, it could be an event that RTC overflows. See chip configuration section for details.

35.4.5 Scan times

The TSI provides multi-scan function. The number of scans is indicated by TSI_GENCS[NSCN] that allow the scan number from 1 to 32. When TSI_GENCS[NSCN] is set to 0 (only once), the single scan is engaged. The 16-bit counter accumulates all scan results until the NSCN time scan completes, and users can read TSI_DATA[TSICNT] to get this accumulation. When DMA transfer is enabled, the counter values can also be read out by DMA engine.

35.4.6 Clock setting

TSI is built with dual oscillator architecture. In normal sensing application, the reference oscillator clock is the only clock source for operations. The reference clock is used to measure the electrode oscillator by ticking a 16-bit counter. The reference oscillator frequency depends on the current source setting. Please refer to the Current source for more details.

The output of electrode oscillator has several prescalers up to 128 indicated by TSI_GENCS[PS]. This allows a flexible counter configuration for different electrode oscillator frequency.

35.4.7 Reference voltage

The TSI module offers a internal reference voltage for both electrode oscillator and reference oscillator. The internal reference voltage can work in low power modes even when the MCU regulator is partially powered down, which is ideally for low-power touch detection.

The charge and discharge difference voltage is configurable upon the setting of TSI_GENCS[DVOLT]. The following table shows the all the delta voltage configurations.

NOTE

This table doesn't apply to noise mode, see noise mode sections for its configuration.

Table 35-11. Delta voltage configuration

DVOLT	V_p (V)	V_m (V)	ΔV (V)
00	1.328	0.302	1.026
01	1.111	0.519	0.592
10	0.986	0.644	0.342
11	0.914	0.716	0.198

35.4.8 Current source

The TSI module supports eight different current source power to increment from 500 nA to 64 µA. TSI_GENCS[EXTCHRG] determines the current of electrode oscillator that charges and discharges external electrodes. The TSI_GENCS[REFCHRG] determines the current of reference oscillator on which the internal reference clock depends. The lower current source takes more time for charge and discharge, which is useful to detect high-accuracy change. The higher current source takes less time, which can be used to charge a big electrode by less power consumption.

TSI_GENCS[CURSW] allows the current source to swap, so that the reference oscillator and electrode oscillator use the opposite current sources. When TSI_GENCS[CURSW] is set and the current sources are swapped, TSI_GENCS[EXTCHRG] and TSI_GENCS[REFCHRG] still control the corresponding current sources, that is, TSI_GENCS[EXTCHRG] controls the reference oscillator current and TSI_GENCS[REFCHRG] controls the electrode oscillator current.

35.4.9 End of scan

As a scan starts, [SCNIP] bit is set to indicate scan is in progress. When the scan completes, the [EOSF] bit is set. Before clearing the [EOSF] bit, the value in TSI_DATA[TSICNT] must be read. If the TSI_GENCS[TSIIEN] and TSI_GENCS[ESOR] are set andTSI_GENCS[DMAEN]is not set, an interrupt is submitted to CPU for post-processing immediately. The interrupt is also optional to wake MCU to execute ISR if it is in low power mode. When DMA function is enabled by setting TSI_GENCS[TSIIEN] and TSI_GENCS[ESOR], as soon as scan completes, a DMA transfer request is asserted to DMA controller for data movement, generally, DMA engine will fetch TSI conversion result from TSI_DATA register,store it to other memory space and then refresh the TSI scan channel index(TSI_DATA[TSICH]) for next loop. When DMA transfer is done, TSI_GENCS[EOSF] is cleared automatically.

35.4.10 Out-of-range interrupt

If enabled, TSI will scan the electrode specified by TSI_DATA[TSICH] as soon as the trigger arrives. The TSI_GENCS[OUTRGF] flag generates a TSI interrupt request if the TSI_GENCS[TSIIE] bit is set and GENCS[ESOR] bit is cleared. With this configuration, after the end-of-electrode scan, the electrode capacitance will be converted and stored to the result register TSI_DATA[TSICNT], the out-of-range interrupt is only requested if there is a considerable capacitance change defined by the TSI_TSHD. For instance, if in low power mode the electrode capacitance does not vary, the out-of-range interrupt does not interrupt the CPU. This interrupt will not happen in noise detection mode. It is worthy to note that when the counter value reaches 0xFFFF is treated as an extreme case the out-of-range will not happen. Also in noise detection mode, the out-of-range will not assert either.

35.4.11 Wake up MCU from low power modes

In low power modes, once enabled by TSI_GENCS[STPE] and TSI_GENCS[TSIIE], TSI can bring MCU out of its low power modes(STOP, VLPS, VLLS,etc) by either end of scan or out of range interrupt, that is, if TSI_GENCS[ESOR] is set, end of scan interrupt is selected and otherwise, out of range is selected.

35.4.12 DMA function support

Transmit by DMA is supported only when TSI_DATA[DMAEN] is set. A DMA transfer request is asserted when all the flags based on TSI_GENCS[ESOR] settings and TSI_GENCS[TSIIE] are set. Then the on-chip DMA controller detects this request and transfers data between memory space and TSI register space. After the data tranfer, DMA DONE is asserted to clear TSI_GENCS[EOSF] automatically. This function is normally used by DMA controller to get the conversion result from TSI_DATA[TSICNT] upon a end-of-scan event and then refresh the channel index(TSI_DATA[TSICH]) for next trigger. DMA function is not available when MCU is in stop modes.

35.4.13 Noise detection mode

The noise detection mode is used to detect power of noise. In this mode the thresholds are incremented internally by TSI until the point that there is no noise voltage trepassing the threshold.

The noise detection mode change the circuit configuration as shown in the following figure. With this configuration, it is possible to detect touch with high levels of EMC noise present. To enter this mode, set TSI_CS3[STAT_STUP] field to 1100b.

In noise detection mode the reference oscillator has the same configuration except the output goes to Counter2 and this counter will have its maximum count set by NSCNx2$^{(PS)}$. This means this oscillator will setup the noise detection mode sense duration.

The blocks of external oscillator is changed and instead of an oscillator the circuit implements an RF amplitude detection. The threshold for this amplitude detection is set by DVOLT register bits.

Also the external voltage is biased by vmid voltage with a Rs series resistance.

The vmid voltage is defined as V(vmid) = (V(vp) + V(vm))/2.

The Rs value is defined by TSI_CS2[EXTCHRG] register bits.

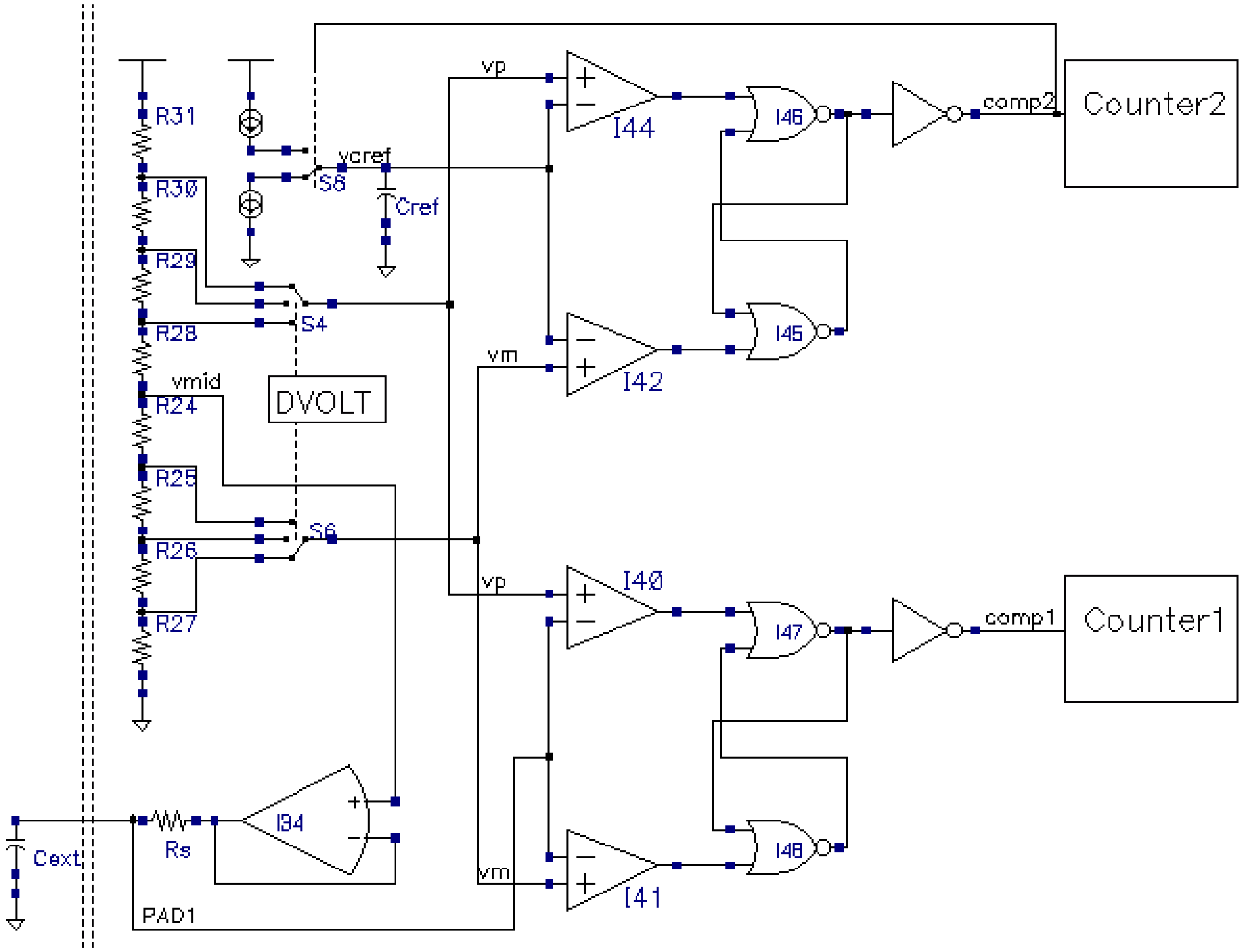


Figure 35-15. TSI circuit in noise detection mode

To determine the noise level the below algorithm can be used:

1. Initialize Rs = maxrs; Dvolt = minDv (set other configurations also)
2. Perform a noise cycle.
3. If TSIcounter < 3, go to step 8
4. If Rs = minrs, go to step 6.
5. Reduce value of Rs. go to step 2
6. If Dvolt = maxDv, go to END
7. Increase value of Dvolt. Set Rs = maxrs. go to step 2
8. If Rs > minrs, reduce value of Rs, go to END
9. Rs = maxrs, reduce value of Dvolt.
10. END Get value of Rs and Dvolt.

One example of noise detection mode is shown in the following figure. in this figure the TSI is working in capacitive mode until 30uS when it is changed to noise detection mode. In noise detection mode the selected pad is biased with 0.815V and all AC waveform in this pad is caused by a noise source external to the MCU.

It is possible to observe in the following figure that, in noise detection mode, the clkref output has the peak detection and the number of detected peaks can be counted or used by digital block. The clkext output has the internal oscillator output and can be used to set the maximum noise detection time window.

The waveform of the following figure shows two operations during noise detection mode:

- The V(vp) and V(vm) thresholds are changed in 34.4 μs.
- The Rs series resistance value is changed between 184 kΩ (TSI_CS2[EXTCHRG]=011b) and 32 kΩ (TSI_CS2[EXTCHRG]=101). Because of this Rs change the amplitude of noise waveform change also.

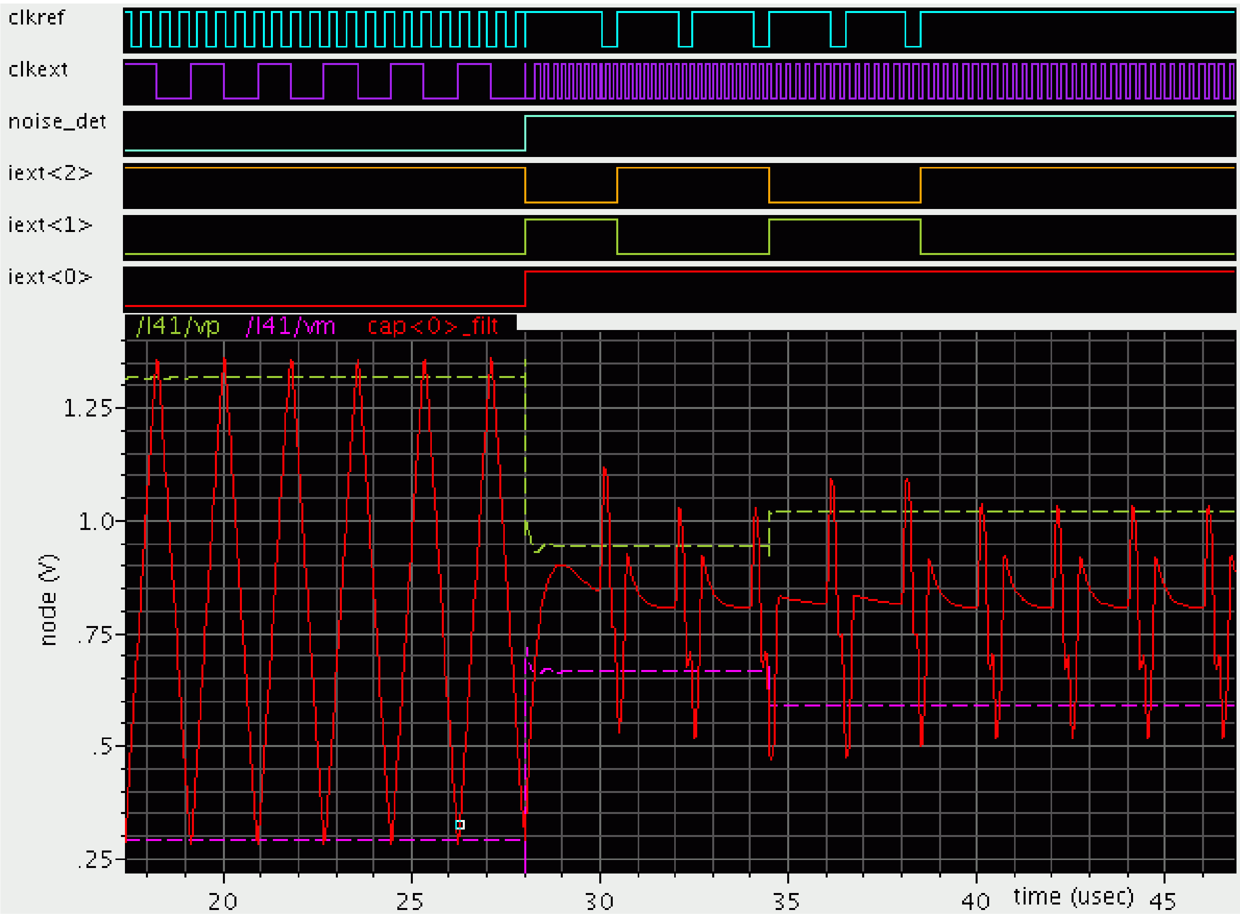


Figure 35-16. TSI noise detection mode waveform

35.4.13.1 Automatic noise mode

This mode is set by MODE[3:2] = 11 (noise mode 3). In this mode, the thresholds are incremented internally by the module until the point that there is no noise voltage trepassing the threshold.

The following diagram shows how it is done. The threshold comparator output goes to a counter and as the DVOLT control bits are increased the DVOLT thresholds are increased as well. The four bits are counted until 1111 (=15) and the counter is stop with this maximum value.

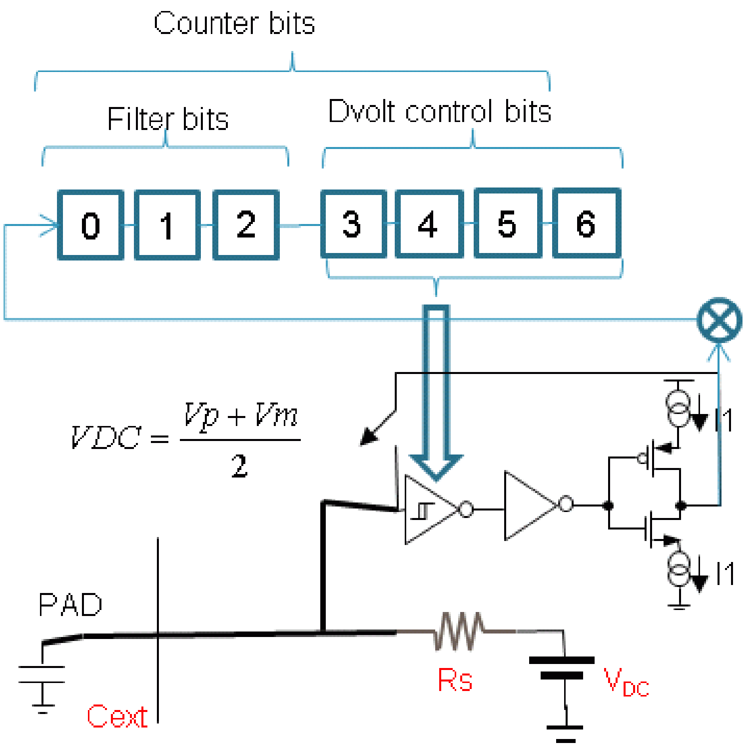


Figure 35-17. Block diagram automatic noise threshold operation

The signals that have different behavior in this noise mode (wrt capacitive mode) are shown in the following table.

Table 35-12. Signal properties in automatic noise operation mode

Name	Function	I/O type	Power Up/Reset state
MODE[3:2]	11—Noise mode operation with frequency limitation and automatic threshold counter.	I	00
EXTCHRG[2:1]	In this operation mode, these bits select the number of filter bits. 00—3 filter bits 01—2 filter bits 10—1 filter bit 11—no filter bit	I	00

Table continues on the next page...

Table 35-12. Signal properties in automatic noise operation mode (continued)

Name	Function	I/O type	Power Up/Reset state
EXTCHRG[0]	In this operation mode, this bit selects the series resistance. 0—uses Rs=32 kΩ 1—uses Rs=187 kΩ Independent of this bit selection, the threshold 15 is done with Rs = 5.5 kΩ	I	0
DVOLT[1:0]	Selects voltage rails of the internal oscillator	I	00
MODE[3:0]	DVOLT counter bits output. This field keeps 0000b if MODE[3:2] is not 11 after entering automatic noise mode.	O	0000

35.4.13.2 Single threshold noise modes

These modes are reset by MODE[3:2]=01 and 10. The difference between these two modes is that in mode 2 (MODE[3:2]=10), there is a frequency limitation circuit that enables the circuit to operate in higher frequencies. In mode 1 (MODE[3:2]=01), this frequency limitation circuit is not enabled.

In this mode, the thresholds are set by user via register bits as described in the following table.

During these modes the internal oscillator rails are set to the maximum (equivalent to DVOLT[1:0]=00).

Table 35-13. Signal properties in single noise modes (1,2)

Name	Function	I/O type	Power up / reset
MODE[3:2]	01 or 10- Single thrshold noise mode operation.	I	00

Table continues on the next page...

Table 35-13. Signal properties in single noise modes (1,2) (continued)

Name	Function	I/O type	Power up / reset
DVOLT[1:0], EXTCHRG[2:1]	In this operation mode these 4 bits are used select the noise threshold. 0000 - DVpm = 0.038 V, Vp = 0.834 V, Vm = 0.796 V 0001 - DVpm = 0.050 V, Vp = 0.830 V, Vm = 0.790 V 0010 - DVpm = 0.066 V, Vp = 0.848 V, Vm = 0.782 V 0011 - DVpm = 0.087 V, Vp = 0.858 V, Vm = 0.772 V 0100 - DVpm = 0.114 V, Vp = 0.872 V, Vm = 0.758 V 0101 - DVpm = 0.150 V, Vp = 0.890 V, Vm = 0.740 V 0110 - DVpm = 0.197 V, Vp = 0.914 V, Vm = 0.716 V 0111 - DVpm = 0.260 V, Vp = 0.945 V, Vm = 0.685 V 1000 - DVpm = 0.342 V, Vp = 0.986 V, Vm = 0.644 V 1001 - DVpm = 0.450 V, Vp = 1.040 V, Vm = 0.590 V 1010 - DVpm = 0.592 V, Vp = 1.111 V, Vm = 0.519 V 1011 - DVpm = 0.780 V, Vp = 1.205 V, Vm = 0.425 V 1100 - DVpm = 1.026 V, Vp = 1.328 V, Vm = 0.302 V 1101 - DVpm = 1.350 V, Vp = 1.490 V, Vm = 0.140 V 1110 - DVpm = 1.630 V, Vp = 1.630 V, Vm = 0 V 1111 - DVpm = 1.630 V, Vp = 1.630 V, Vm = 0 V	I	XXXX
EXTCHRG[0]	In this operation mode this bits selects the series resistance. 0 - uses Rs = 32 kΩ. 1- uses Rs = 187 kΩ. Independent of this bit selection the threshold 15 is done with Rs = 5.5 kΩ.	I	XX